miesięcznik profesjonalistów

10
1998

informatyka

Cena: 6 zł. ISSN 0542 9951 WYDAWNICTWO SIGMA NOT

P 1877/98

Pismo informatyczne - ukazuje się od 1965 roku

Pomiar niezawodności oprogramowania

Inspekcje oprogramowania

Problemy z utrzymaniem dużych baz danych

Zaminowany teren dużych systemów komputerowych

diament

w doskonałej oprawie

Współpraca tworzy nową wartość

Softbank S.A. we współpracy z brytyjską firmą GFS przedstawia nowego aktora na scenie informatyki bankowej. Diamond – scentralizowany system najnowszej generacji.

Architektura klient–serwer, technologia obiektowa, otwartość oraz możliwość pracy na wielu platformach tworzą unikatowy zestaw cech systemu dla banku, pragnącego sprostać wyzwaniom przyszłości.

Dzięki nim można system adaptować do wymagań użytkownika a nie odwrotnie.

Technologia informatyczna to nie wszystko – jest jeszcze funkcjonalność. Diamond zapewnia obsługę działalności detalicznej i hurtowej banku w całości lub jej wybranych obszarów – we współpracy z innymi systemami.

Klejnot wymaga oprawy. Systemowi Diamond daje ją Softbank S.A. – firma o ugruntowanej pozycji lidera rozwiązań informatycznych dla polskiej bankowości.

Diament w doskonałej oprawie

SOFTBANK® S.A.

ul. 17 Stycznia 72a, 02-146 Warszawa, tel. (022) 878 62 00, fax (022) 878 63 00

GFS Diamond

informatyka

Miesięcznik
ISSN 0542-9951
INDEKS 36124

P1877/98

POLITECHNIKA ŚLĄSKA Biblioteka Główna w Gliwicach

nr 10 1998
Październik
Rok wydania XXXIV

REDAGUJE ZESPÓŁ:
dr Lesław **WAWRZONEK**
(redaktor naczelny)
Alina **KLEPACZ**
(sekretarz redakcji)
redaktorzy:
mgr Zdzisław **ŻURAKOWSKI**
dr Ewa **ŁUKASIK**
współpraca:
Rafał **MAŚLANA**
Ewa **DULNA** (korekta)
Kamila **PODRECKA** (adm. red.)

KOLEGIUM REDAKCYJNE:
prof.dr hab. Leonard BOLC
mgr inż. Piotr FUGLEWICZ
prof. dr hab. Jan GOLIŃSKI
dr inż. Zenon KULPA
prof. dr inż. Jan MULAWKA
prof. dr hab. Wojciech OLEJNICZAK
mgr. inż. Jan RYŻKO
dr Witold STANISZKIS
dr inż. Jacek STOCHLAK
prof. dr hab. Maciej STOLARSKI
prof. dr hab. Zdzisław SZYJEWSKI
prof. dr hab. inż.
Ryszard TADEUSIEWICZ
prof. dr hab. Jan WĘGLARZ
PRZEWODNICZĄCY
RADY PROGRAMOWEJ
prof. dr hab. Juliusz Lech KULIKOWSKI

WYDAWCA:
Wydawnictwo Czasopism i Książek Technicznych SIGMA NOT Spółka z o.o.
ul. Ratuszowa 11
00-950 WARSZAWA
skrytka pocztowa 1004

REDAKCJA:
00-950 Warszawa,
ul. Ratuszowa 11, p. 644, 628
skrytka pocztowa 1004
tel., fax: 619-11-61
tel.: 619-22-41 w. 159
e-mail: informat@pol.pl
www.pol.pl/informatyka/

PRENUMERATA:
tel. 40-35-89, 40-30-86
Prenumeratę przyjmujemy również w sieci INTERNET:
WWW.pol.pl./sigma_not
E-mail: kolpor.sigma@pol.pl

Materiałów nie zamówionych redakcja nie zwraca.
Autorzy artykułów proszeni są o przysyłanie tekstów na dyskietkach 3½" lub pocztą elektroniczną - w edytorach Word.
Redakcja zastrzega sobie prawo dokonywania zmian w nadsyłanych materiałach.
Po szczegółowe informacje dla Autorów prosimy zwracać się do redakcji.
Redakcja nie ingeruje w treść i formę ogłoszeń oraz innych materiałów reklamowych, w związku z tym nie ponosi za nie odpowiedzialności.

Ogłoszenia przyjmują:
- Redakcja, tel. 619-11-61
- Dział Reklamy i Marketingu
00-950 Warszawa, ul. Mazowiecka 12
tel.: 827-43-66, fax: 826-80-16

Okładka:
AGAT, Jerzy Burski i Andrzej Jacyszyn
Łamanie:
Alina Klepacz, program PageMaker

Druk:
Drukarnia SIGMA NOT Sp. z o.o.

W numerze:

SUN tylko dla dorosłych - Lesław Wawrzonek 2

INFORMACJE 3

WYWIAD

Recepta na sukces – inwestuj zyski 7
Rozmowa z dr. Dieterem Küchlerem, prezesem firmy LIPRO GmbH

PUBLIKACJE

Problemy z utrzymaniem dużych baz danych 10
Wojciech Czujowski, Jacek Gruber

Pomiar niezawodności oprogramowania 17
Piotr Jędrzejowicz

Podstawowe elementy krytyki reengineeringu i analiza ich zasadności 21
Janusz Wielki

Miary oprogramowania 26
Andrzej Kobyliński

Dane hurtowo 30
Robert Wrembel

Inspekcje oprogramowania 36
Janusz Górski

OPINIE

Zaminowany teren dużych systemów komputerowych 39
Jacek Kryt

BIULETYN PTI 43

Uprzejmie informujemy, że począwszy od września br. w Internecie będą zamieszczane spisy treści kolejnych zeszytów czasopism, wydawanych przez Wydawnictwo SIGMA-NOT.

Sun **tylko dla dorosłych**

Targi technik komputerowych i doniesienia prasy o natrętnej jak muchy młodzieży są tak nierozłączne, jak pisanki w TV i święta wielkanocne. Dzieciarnia oczyszcza stoiska ze wszystkich ruchomości - ulotek, długopisów i breloczków, a gdy „zdolniejsza", także i z portfeli - piszą z obrzydzeniem ci, którzy sami jeszcze kilka lat temu, nie schylając się, przechodzili pod targowymi barierkami. No cóż, nagłe przejście do grupy starszaków może zamieszać w głowie.

Oryginalną taktyką obronną przed plagą małolatów zastosowały na Softarg'98 firmy – partnerzy Suna. Na drzwiach swojego pawilonu umieściły one napis: *Młodzież zapraszamy po godzinie 15.00* (targi trwały do 17.00). Dla większej pewności wejścia pilnowała panienka-ochroniarz, która klasyfikowała w skali zerojedynkowej wiek natrętów, ciekawych oferty firm „za zamkniętymi drzwiami". Szczęśliwie (w tym wypadku), do młodzieży już się nie zaliczam, więc sprawdziłem co też ciekawego, specjalnie dla dorosłych, przygotował Sun. Może będą roznegliżowane Javajki? Niestety, nic z tych rzeczy. Jedyną nieprzyzwoitością był sposób potraktowania zwiedzających, którzy przecież zapłacili za bilety do wszystkich pawilonów.

Zupełnie inne podejście do młodzieży pokazał, nie po raz pierwszy, Microsoft. Przeprowadził on swoistą „Akcję małolat". Polegała ona na zorganizowaniu serii prezentacji i wykładów o Windows'98 PL, a uczestniczyło w nich w sumie około 1500, głównie młodych, słuchaczy. Kawał dobrej edukacyjnej i ... marketingowej roboty. *To, że jesteśmy monopolistami, zawdzięczamy konkurencji* - powiedział mi Jacek Myrcha z Microsoftu - wskazując na wspomniany pawilon „dla dorosłych". Trudno odmówić mu racji.

Organizatorzy CeBIT-u problem młodzieży rozwiązują, stosując prostą zasadę: przez kieszeń do rozumu. Według niej, tylko w ostatnim dniu Targów, a także w niedzielę, młodzież może kupić bilety ulgowe, dwa i pół razy tańsze niż normalne. W ten niemal naturalny sposób oddziela się dni dla specjalistów i handlowców od dni dla młodzieży. Innych różnic w traktowaniu przyszłych i aktualnych klientów na CeBIT nie ma. Podobnie jest na targach Invex w Brnie, które odwiedzają nieprzebrane tłumy uczniów i studentów.

Dla naszych wystawców idealnym gościem targów jest kontrahent podpisujący na stoisku umowę na zakup systemu, programów lub komputerów. Wizyty pozostałych to strata czasu, ulotki lub długopisu. Marzą im się targi, na które bramkarze wpuszczają tylko tych, którzy zapewnią, że przyszli tu, aby podpisać kontrakt chociaż z jednym z wystawców. W przeciwnym razie usłyszą: wynocha, paszoł won, raus, clear out - jak na międzynarodowe targi przystało.

Odgrodzeni od szarej gawiedzi szybami i hostessami, pracownicy firm, absolwenci wyższych uczelni, zachowują się jak uciekinierzy, którzy po wdrapaniu się po drabinie, wciągają ją za sobą. Myślą, że skoro mają już tytuły i doktoraty, to nie są już potrzebne wyższe uczelnie, czasopisma naukowo-techniczne i targi z ich edukacyjną funkcją. Konkurencję - młodsze pokolenie - najlepiej zwalczać, blokując jej dostęp do wiedzy i wykształcenia. Oni, starsi i mądrzejsi potrafili, *znaleźć się w nowej rzeczywistości*, jak mówi dr Syryjczyk. Do osiągnięcia pełni sukcesu wystarczy już tylko opanowanie umiejętności: zasiadania, zagajania, przewodniczenia i uświetniania.

Bardzo szybko zapomnieli, w jaki sposób dochodzili do swojej aktualnej pozycji. Zaczynali przecież od czytania, dogorywającego dziś, „Młodego Technika" i nieistniejących już „Horyzontów Techniki", oglądania filmów popularnonaukowych w Muzeum Techniki czy słuchania wykładów Uniwersytetu Powszechnego. Uczestniczyli w wycieczkach na przykład do Instytutu Badań Jądrowych, przypominającego dziś miasteczko poszukiwaczy złota po wyczerpaniu okolicznych zasobów tego kruszcu. Zwiedzali, zlikwidowane już zakłady CEMI wytwarzające materiały dla zaawansowanej elektroniki, ELWRO - produkujące duże komputery, czy zabrzański ELZAB, producenta minikomputerów klasy PDP/11 (teraz kasy fiskalne). Dziś najbliższa fabryka dużych komputerów jest w Montpellier na południu Francji, a nowoczesne technologie przeniosły się z polskich instytutów i fabryk do biur przedstawicieli handlowych. Co zobaczy młodzież zainteresowana nowoczesną techniką w biurach Novella, Oracle'a czy Intela? To, co w każdym biurze.

Jedyną możliwością kontaktu młodych ludzi z nowoczesną techniką są właśnie targi. Na innych niż komputerowe nie istnieje problem zaludniania (to też ludzie) przez małolatów hal wystawowych. Nie pasjonują ich maszyny piekarnicze, budowlane czy dziewiarskie. Tylko po pokazie produktów informatycznych można zobaczyć chłopca, który z wypiekami na twarzy mówi: *Mamo, Ty wiesz, co ten system potrafi!!!*. Powszechne zainteresowanie młodzieży komputerami powinno chyba cieszyć a nie drażnić pracowników firm branży informatycznej (przyszli klienci i pracownicy), jak i dostawców tekstów dla pism komputerowych (aktualni czytelnicy).

Sposób docierania do młodzieży szkolnej i studentów wiele mówi o tym, jak firma widzi swoją przyszłość w naszym kraju. Większość z nich działa tak, jakby nie wierzyły, że doczekają czasów, kiedy obecna młodzież stanie się ich klientami lub pracownikami. Po co zresztą uczestniczyć w targach, skoro można zaprosić na Majorkę zarząd, na przykład banku, aby w pięknych okolicznościach przyrody przetestował proponowane oprogramowanie. Kosztuje to znacznie mniej niż stoisko wystawowe, a do tego jest to pewniejsza inwestycja. No i najważniejsze - nie ma tych okropnych dzieciaków.

Lesław Wawrowski

Alcatel współpracuje z Politechniką Warszawską

Zgodnie z podpisanym porozumieniem Alcatel Polska i Politechnika Warszawska będą współpracować w zakresie prac badawczo-rozwojowych, między innymi poprzez wspólne określanie tematów badań naukowych i opracowań technicznych. Istotnymi elementami współpracy będą także: modernizacja zaplecza laboratoryjnego instytutów politechnicznych, kształcenie i doskonalenie zawodowe kadr Alcatela oraz praktyki dla studentów.

Kontrakt Wawela z JBA

Pod koniec czerwca br. Zakłady Przemysłu Cukierniczego WAWEL S.A. w Krakowie podpisały kontrakt z JBA Polska – Oddział Katowice na dostawę zintegrowanego systemu zarządzania przedsiębiorstwem. ZPC WAWEL S.A. Platformą dla nowego systemu będzie komputer AS/400e. Umowa uwzględnia 3–letni okres usług wsparcia wdrożenia oraz wieloletni serwis. Zakończenie wdrożenia systemu przewidziano na marzec 1999 roku. W systemie będzie pracowało 100 końcówek, które obejmą wszystkie wydziały zakładu i zachodzące w nich procesy.

MIS został Premium Lotus Buisness Partner

W lipcu tego roku wrocławska firma Management Information Systems Sp. z o.o. dołączyła do grona Premium Lotus Buisness Partner. Jest to najwyższy tytuł nadawany partnerom przez firmę Lotus. W Polsce są tylko trzy firmy posiadające taki status.

Solidex S.A. dla British Petroleum

Solidex wygrał przetarg i rozpoczął realizację dwóch projektów dla British Petroleum. Pierwszy z nich obejmuje realizację sieci rozległej BP na podstawie routerów Cisco. Sieć ta połączy wszystkie biura BP zlokalizowane w kilkunastu miejscowościach rozsianych po całym kraju, umożliwiając im korzystanie ze wspólnych zasobów centrali firmy. Sieć ta objęta zostanie specjalizowanym serwisem gwarancyjnym obejmującym bardzo krótkie czasy reakcji i naprawy urządzeń krytycznych dla jej funkcjonowania. Drugim projektem jest dostawa, instalacja i serwis serwera z bazą danych. Serwer ten, w połączeniu z siecią komputerową, będzie stanowił podstawowy system księgowy dla BP.

Sieć ATM dla Zakładów Chemicznych „Rokita" S.A.

Projekt swoim zakresem obejmuje budowę sieci kampusowej opartej na technologii ATM wraz z węzłami dostępowymi oraz budowę korporacyjnego Intranetu i węzła sieciowego „ROKITA" S.A. Projekt sieci szkieletowej i węzłów dostępowych zakłada wysoki stopień redundancji i odporności na ewentualne awarie, niezależnie od zastosowania nowoczesnych rozwiązań do monitoringu i zdalnego zarządzania siecią. Przetarg na realizację projektu sieci wygrał Solidex S.A.

Internetowe mieszkania

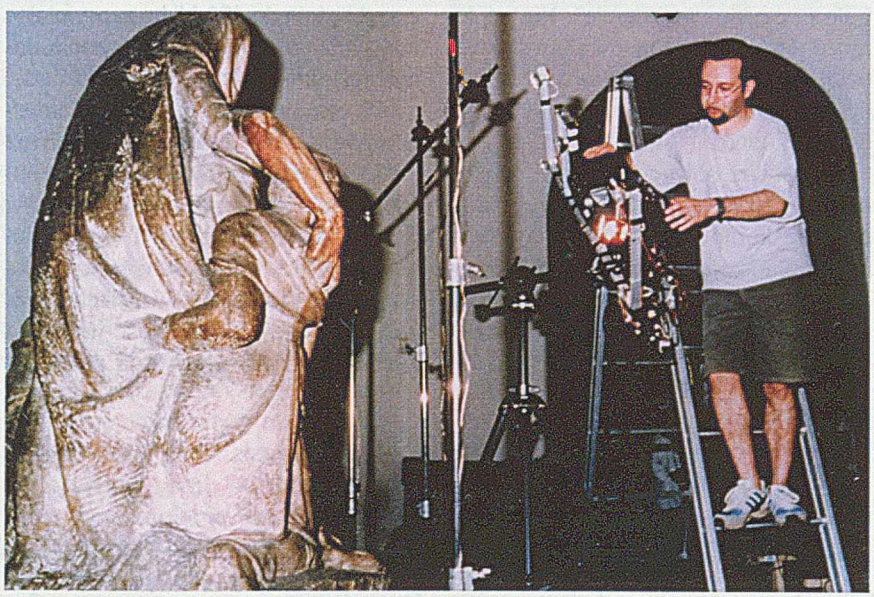
Spółdzielnia Mieszkaniowo-Budowlana „Dembud", we współpracy z Polską OnLine, wyposażyła wszystkie lokale wybudowane przy ulicy Łuckiej 18 i 20 w Warszawie w lokalną sieć komputerową, zgodną ze standardem Ethernet. Zapewnia ona transmisję danych z szybkością 10 MB/s. Dla każdego mieszkania w spółdzielni przewidziano adres e-mail, 5 MB na serwerze oraz dostęp do wszystkich usług Internetu: WWW, ftp, telnet, irc. Użytkownicy nie są rozliczani z czasu pracy w sieci, ani z ilości przesyłanych danych.

QAD Polska

Firma QAD producent systemu do zarządzenia przedsiębiorstwem – MFG/PRO otwiera oddział w Polsce. W pierwszym etapie powstaną trzy biura w Pol-

Odkrywanie tajemnic Piety Florenckiej

Badania nad Pietą Michała Anioła stanowią przedmiot współpracy pomiędzy grupą specjalistów z dziedziny informatyki z TJ Watson Research Center, a historykiem sztuki epoki renesansu, Jackiem Wassermanem. Rozwiązania IBM pozwolą na stworzenie prawie doskonałego trójwymiarowego cyfrowego modelu Piety Florenckiej. Będzie to wymagało zgromadzenia około dwóch miliardów bitów danych. Badania te mają rzucić nowe światło na tajemnicę Piety Florenckiej, a jednocześnie stać się podstawą nowych technologii i zastosowań trójwymiarowych modeli komputerowych rzeczywistych obiektów. W rezultacie powstanie wirtualny model Piety, zachowany w pamięci komputera i umożliwiający wykonanie badań, niemożliwych do przeprowadzenia na oryginale.



sce – we Wrocławiu, Warszawie i na Wybrzeżu. W najbliższym czasie firma planuje zatrudnienie około 50 pracowników. Dotychczas pakiet ten wdrażał CSBI.

20 lat na pierwszym miejscu

Według corocznego rankingu sporządzanego przez magazyn „Electronic Business" - IBM już od dwudziestu lat zajmuje pierwszą pozycję na liście „Top 200 Electronics Companies" – najlepszych firm z sektora przemysłu elektronicznego. Jedynie IBM i Xerox nieprzerwanie od początku trwania rankingu były klasyfikowane wśród pierwszej dziesiątki liderów. Według najnowszych wyników, uwzględniających przychody firm z tytułu sprzedaży produktów elektronicznych, pierwszych dziesięć miejsc zajmują: IBM, Hewlett-Packard, Motorola, Lucent Technologies, Intel, Compaq, Ingram Micro, EDS, Xerox i Microsoft.

Polskie Windows 98

7 września br. Microsoft pokazał po raz pierwszy polską wersję nowego systemu operacyjnego Windows 98. Interfejs z poprzedniej wersji nie różni się zasadniczo od nowego, ale do systemu wprowadzono ponad 3000 poprawek. Dodano obsługę nowych urządzeń zewnętrznych, urządzeń wyposażonych w łącze USB, czytników DVD. Dołączono obsługę FAT32. Sztandarową zmianą w stosunku do Windows 95 jest integracja z Internetem, który „zajął centralną pozycję w systemie". Nie są to może rewolucyjne zmiany, ale w trakcie użytkowania odnosi się wrażenie, że system jest zdecydowanie stabilniejszy od poprzednika.

Sprzedaż nowego systemu rozpoczęła się 28 września. Od tej pory sprzedawane komputery powinny już być wyposażane w nową wersję Windows. Jeżeli mamy już komputer, to za wersję „upgrade" zapłacimy 340 zł, a za pełną wersję 570 zł.

e-czeki

IBM bierze udział we współpracy nad projektem mającym umożliwić regulowanie płatności czekami poprzez Internet. W rezultacie banki będą mogły zaoferować swoim klientom sprawną metodę wystawiania i przyjmowania czeków za pośrednictwem sieci. e-czeki podlegają tym samym procedurom i przepisom co czeki papierowe, umożliwiają zatem bezpieczne dokonywanie płatności za pośrednictwem Internetu. Stanowią one prosty sposób włączenia się do handlu elektronicznego, bez potrzeby inwestowania w nowe technologie czy wyspecjalizowane systemy.

Compaq i Microsoft dla Unixa i NT

Compaq nawiązał współpracę z Microsoftem w celu usprawnienia współdziałania systemu Digital UNIX Compaqa z systemami Windows NT Server i Windows oraz zmniejszenia złożoności i ryzyka pracy w środowiskach mieszanych. Obie firmy postanowiły również sukcesywnie wprowadzać funkcje niezawodnego systemu operacyjnego NonStopTM Kernel do przyszłych wersji Windows NT Servera.

Planuje się stopniowe instalowanie coraz większej liczby aplikacji w bardziej niezawodnych, w przyszłych wersjach Windows NT. Zapowiedź współpracy z 64-bitowym systemem Windows NT jest bardzo ważnym czynnikiem dla klientów, którzy dokonują obecnie wyboru systemu operacyjnego.

Nowy standard PCI

Firmy Compaq, HP i IBM współpracują nad nową technologią magistrali PCI. 64-bitowa magistrala PCI-X działa z szybkością do 133 MHz, zapewniając maksymalną szybkość przesyłania danych przekraczającą 1 GB na sekundę, osiem razy wyższą niż najpopularniejsze obecnie zastosowania PCI. Wraz z PCI-X wprowadzany jest również ulepszony protokół, usprawniający przepływ danych i upraszczający wymagania dotyczące taktowania, ważnego czynnika przy wyższych częstotliwościach zegara. PCI-X utrzyma zgodność wsteczną z aktualnie wykorzystywaną magistralą PCI.

ECIS 2002 w Gdańsku

Stały komitet konferencji ECIS (*European Conference on Information Systems*) powierzył przewodnictwo komitetowi organizacyjnemu ECIS 2002 profesorowi Stanisławowi Wryczy, prezesowi Naukowego Towarzystwa Informatyki Ekonomicznej i kierownikowi Katedry Informatyki Ekonomicznej Uniwersytetu Gdańskiego. Stroną techniczno-organizacyjną konferencji zajmą się Międzynarodowe Targi Gdańskie.

Nasz człowiek w CEPIS

Piotr Fuglewicz, wiceprezes PTI, członek rady programowej „Informatyki" został wiceprzewodniczącym Rady Europejskich Profesjonalnych Stowarzyszeń Informatycznych (CEPIS - *Community of European Professional Informatics Societies*). Rada powierzyła nowemu wiceprzewodniczącemu zakres obowiązków obejmujący sprawy rozwoju i promocji pojęcia profesjonalizmu informatycznego w skali europejskiej. (szczegóły w biuletynie PTI, 10/98).

Windows NT 4.0 szybsze od Netware 5.0

Niezależna amerykańska firma Mindcraft, specjalizująca się m.in. w badaniach wydajnościowych oprogramowania, przeprowadziła testy porównawcze Windows NT Server 4.0 oraz Netware 5.0. W wyniku testów okazało się, że Windows NT 4.0 jest szybszy o 25% od swego rywala oraz 2,7 razy wydajniejszy jako serwer plików. Wyniki dotyczą zarówno pracy na maszynie z jednym, jak i dwoma procesorami. Więcej informacji na ten temat można znaleźć na stronie: http://www.mindcraft.com/whitepapers/

Informatyka dla samorządów - konferencja

Polskie Towarzystwo Informatyczne wraz z Fundacją Rozwoju Demokracji Lokalnej, Ośrodkiem Kształcenia Samorządu Terytorialnego im. Waleriana Pańki w Katowicach organizują w dniach 4 - 6 listopada br. kolejną, trzecią konferencję na temat „Informatyka w administracji samorządowej".

Na konferencji będą prezentowane oferty firm informatycznych w zakresie rozwiązań dotyczących szeroko rozumianej administracji, nie tylko jak do-

tychczas urzędów miejskich i gminnych, ale także urzędów powiatowych, wojewódzkich i jednostek im podległych, jak na przykład Ośrodków Pomocy Społecznej. Firmy będą miały do dyspozycji odrębne sale, w których mogą prowadzić pokazy oprogramowania. Będą także prelekcje, poświęcone zarówno informatyce, jak i innym tematom ważnym dla środowiska, np. reformie samorządowej. Zaproszeni zostali twórcy reformy do wygłoszenia wykładów na temat funkcjonowania przyszłych powiatów i województw.

Konferencja odbędzie się w Ustroniu k. Bielska-Białej. Szczegóły podaje biuro Fundacji w Katowicach tel./fax (032) 206 98 43, tel. 253 84 09.

Nagrody dla najlepszych w Europie innowacyjnych firm informatycznych

Ogłoszono listę 25 tegorocznych laureatów Europejskiej Nagrody Informatycznej – ustanowionej w celu stymulowania innowacji w dziedzinie informatyki w Europie – na którą została przeznaczona łączna kwota 725.000 ECU (około 2 755 000 PLN). Program ten jest koordynowany przez Komisję Europejską wspólnie z Europejską Radą Badań Stosowanych i Technologii Euro-CASE.

Wyróżniono 25 instytucji spośród około 300 z całej Europy. Każdy wyróżniony otrzyma 5 000 ECU Nagrody (w przybliżeniu 19 000 PLN) oraz będzie miał szansę zostać jednym z trzech zwycięzców Nagrody Głównej i otrzymać 200 000 ECU (około 760 000 PLN).

Pośród zwycięzców jest firma z Polski: **„QWED" s.c.** i jej produkt QuickWave-3D software package (Symulator elektromagnetyczny – oprogramowanie do analizy i przemysłowego projektowania urządzeń wysokich częstotliwości.

Nagrodzone produkty obejmują szeroki zakres zastosowań - od kompleksowych, bezpiecznych rozwiązań dla handlu elektronicznego do najnowocześniejszych systemów sprzętowo-programowych do trójwymiarowego modelowania i symulacji szkoleniowych, poprzez aplikacje o silnym społecznym wkładzie w wizualizację dla potrzeb medycyny i kontrolę środowiska.

Nominacja zwycięzców jest wynikiem selekcji wykonanej przez niezależnych ekspertów z 16 krajów europejskich. Zwycięzcy Nagrody Głównej zostaną wybrani przez Jury Wykonawcze składające się z powszechnie znanych europejskich osobistości reprezentujących naukę, technikę i biznes.

Europejska Nagroda Informatyczna ma na celu podniesienie poziomu europejskiej innowacyjności w dziedzinie informatyki, która zdominowana jest przez Amerykę Północną, region Azji i Pacyfiku.

Przewodniczący Komisji Europejskiej Jacques Santer powiedział: *Europa ma bardzo dużo do zaoferowania w zakresie przemysłu informatycznego i mamy nadzieję, że dzięki tej nagrodzie zostanie odkrytych jeszcze więcej talentów z Europy. W tym roku mieliśmy ogromny odzew na ten konkurs i czujemy się podbudowani wysoką jakością zgłoszonych produktów.*

Więcej informacji na stronie: http://www.it-prize.org

Pomiary i zarządzanie IT według Gartner Group

We wrześniu br. odbyło się kolejne seminarium Gartner Group (GG). Tym razem do Warszawy przyjechał Len Bergstrom, wiceprezes GG i założyciel IT Measuerement Services w GG. Wygłosił on dwa wykłady na temat pomiarów i zarządzania całkowitym kosztem posiadania zasobów informatycznych TCO (*Total Cost of Ownership*) oraz pomiarów i zarządzania technologiami informatycznymi IT. A oto w skrócie ważniejsze tezy obu prelekcji:

IT – pomiary i zarządzanie

Nadszedł czas, aby przemysł zaczął postrzegać technologie informatyczne jako „biznes w biznesie". IT przestaje być oceniane w kategoriach sztuki, gdzie sukces jest rzeczą gustu czy nauki, a decyduje o nim garstka wtajemniczonych. Obecnie zaczyna się wymagać dokładnych pomiarów i porównań z najlepszymi. Aby temu sprostać główni informatycy muszą zacząć patrzeć na IT jak na biznes, nieustannie szukając coraz skuteczniejszych sposobów działania. Doświadczenie pokazuje, że *benchmarking* – mierzenie i porównywanie ze sobą środowisk operacyjnych – jest podstawą nieustannych innowacji w dziedzinie IT. Skoro w ciągu najbliższych pięciu lat menadżerowie IT będą koncentrowali się na zmniejszaniu strat i podnoszeniu efektywności, to pomiary staną się kluczowym składnikiem procesów IT.

Pomiary spełniają rolę nawigacji na drodze przemian technologicznych, wskazują obecną pozycję i kierunek w którym się podąża. Pozwalają one na uzasadnienie podejmowanych kroków, dają odpowiedzi na pytania: dlaczego to robimy? gdzie zaczęliśmy? co chcemy osiągnąć? gdzie jesteśmy teraz? Każda firma jest inna, ale zawsze można znaleźć kogoś, z kim warto się porównywać w danej dziedzinie. Mogą to być firmy zajmujące się podobną działalnością lub posiadające zbliżoną liczbę pracowników. Można porównywać się z samym sobą: rezultaty z oczekiwaniami, stan sprzed roku ze stanem obecnym. Wreszcie można rozważyć alternatywy na rynku usług firm zewnętrznych – może nie opłaca się tego robić samemu.

Według GG (prawdopodobieństwo 0.8) przetrwają tylko te działy informatyki, które potrafią zrównoważyć dziedzictwo przeszłości z innowacjami technologicznymi. Zarówno zbyt duży nacisk na wprowadzanie nowych technologii, jak i zbytnie przywiązanie do starych systemów, są drogami skazanymi na niepowodzenie.

W przeszłości przedsiębiorstwa domagały się od swoich działów informatyki kierowania się wydajnością. Wynikało to głównie z nacisku na obniżanie kosztów i z faktu, że wydajność jest łatwa do zmierzenia. Jednak koszt nie zawsze jest najlepszym doradcą. Spójrzmy na ulice – dlaczego wszyscy nie jeżdżą maluchami? Może niska cena to nie wszystko, czego oczekujemy od samochodu. Podobnie jest z przedsięwzięciami w dziedzinie IT. Obecnie od departamentów informatyki oczekuje się zwiększania efektywności. Wymaga to pomiarów i porównań uwzględniających nie tylko koszty, ale również elementy strategiczne i taktyczne, analizy ilościowe i jakościowe. GG przewiduje (prawdopodobieństwo 0.8), że do roku 2000 ponad połowa działów informatyki zmieni swój sposób pracy ze zmniejszania kosztów operacyjnych (wydajność) na zwiększanie wartości biznesowej (efektywność).

Pierwsze pomiary są drogie. Jeśli jednak zacznie się je przeprowadzać odpowiednio często, staną się one naturalnym elementem pracy firmy. Celem tych pomiarów nie powinno być uzyskanie dobrych wyników, aby uspokoić kierownictwo, ale dokładne określenie zmian, które nastąpiły, aby umożliwić wytyczenie dalszej drogi. GG ocenia, że w ciągu pięciu lat podwoi się liczba dużych przedsiębiorstw nieustannie biorących udział w testach i porównaniach w dziedzinie IT (prawdopodobieństwo 0.7).

Całkowity koszt użytkowania - pomiary i zarządzanie

Dokładne pomiary całkowitego kosztu użytkowania (TCO - *Total Cost of Ownership*) w wypadku rozproszonych środowisk komputerowych stały się kluczowym narzędziem do dobrego zarządzania tymi gwałtownie rosnącymi i zmieniającymi się środowiskami. Pomiary TCO dostarczają departamentom informatyki informacji o kosztach i o możliwych oszczędnościach.

W miarę pojawiania się coraz nowszych rozproszonych technologii potrzeba coraz więcej pracy i umiejętności, aby nimi zarządzać. Często, kiedy przedsiębiorstwa modernizują swoje systemy informatyczne, koszty przestają być jasno widoczne, a stają się trudniejszymi do zrozumienia kosztami ukrytymi. Środowiska przetwarzania rozproszonego stają się coraz bardziej złożone i coraz więcej krytycznych dla całego przedsiębiorstwa aplikacji opiera na nich swoje działanie. Te czynniki przyczyniają się do wzrostu kosztów i powodują, że efektywne zarządzanie TCO stało się jednym z najważniejszych problemów kierowników działów IT. Z ostatnich badań wynika, że redukcja TCO jest drugim najważniejszym zagadnieniem IT (zaraz za rokiem 2000) wśród czołowych przedsiębiorstw.

Główne elementy składowe TCO to:

- kapitał;
- administracja;
- wsparcie techniczne;
- operacje użytkowników końcowych.

Według GG przedsiębiorstwa, które nie będą brały pod uwagę wszystkich kosztów, nie zauważą wzrostu kosztów IT powyżej usprawiedliwionego poziomu i będą podejmowały złe decyzje w tej dziedzinie (prawdopodobieństwo 0.9). Dla przykładu obliczenia w modelu klient/serwer wymagają trzykrotnie więcej zasobów wsparcia technicznego niż porównywalne aplikacje w modelu centralnym, ale tylko nieliczne przedsiębiorstwa uwzględniają ten wzrost jako część projektu. W wypadku Windows 95 stosunek pracy do kapitału wynosi: 70% praca, 30% kapitał.

Aby obniżyć koszty związane z TCO, można zrobić wiele, jednak najpierw należy zdawać sobie sprawę z tego, co powoduje ich wzrost. Do metod obniżania kosztów należą: zmniejszanie złożoności, standaryzacja, opracowanie strategii wsparcia – podział na lokalne i centralne, wprowadzanie zaawansowanych technik szkolenia, redukcja liczby dostawców zewnętrznych i uproszczenie kontraktów.

Prelekcję przygotowała firma Hogart z Warszawy, reprezentująca Gartner Group w Polsce.

opr. Rafał Maślana

Recepta na sukces – inwestuj zyski

Z dr. Dieterem Küchlerem, prezesem firmy LIPRO GmbH, rozmawia Lesław Wawrzonek.

L.W.: *Rozwój i produkcja oprogramowania wspomagającego zarządzanie przedsiębiorstwami to chyba jedyna sfera oprogramowania użytkowego, która nie jest całkiem zdominowana przez amerykański przemysł software'owy. Dlaczego akurat ta dziedzina?*

D.K.: Moim zdaniem są dwie przyczyny. Pierwsza to fakt, że logistyka, a później oprogramowanie dla niej, powstały w Europie, a dokładniej w Niemczech, około 20 lat wcześniej niż w USA. A więc tradycja i wyprzedzenie. Po drugie, wielcy producenci oprogramowania, jak na przykład SAP, nadają tempo temu przemysłowi i dlatego w Europie mamy wielu specjalistów z tej dziedziny, podczas gdy w USA jest ich relatywnie mniej. Ten stan ma także złe strony. Europejski problem polega na tym, że jest zbyt wiele starych, dobrych rozwiązań. Kiedy odnosi się sukcesy, nie odczuwa się potrzeby zmian i innowacji. W rezultacie doszło do tego, że ciekawe i nowatorskie rozwiązania w dziedzinie oprogramowania dla przedsiębiorstw powstają ostatnio w USA. Samozadowolenie jest niebezpieczne w każdym biznesie i trzeba się go wystrzegać.

L.W.: *Duże sukcesy obniżyły więc dynamikę rozwoju i Wy na tym skorzystaliście.*

D.K.: Można tak powiedzieć. Kiedy zaczęliśmy szukać dla siebie nowych możliwości, był początek lat 90. Razem z trzema osobami, z którymi realizowaliśmy dość standardowy projekt informatyczny dla przedsiębiorstwa, zauważyliśmy, że ten sam cel można osiągnąć dużo bardziej efektywnie i z większą korzyścią dla zlecającego. Opracowaliśmy odpowiedni prototyp z teoretykami z Instytutu Maszyn i Konstrukcji im. Fraunhofera oraz z Siemens Nixdorf Informationsysteme (SNI). Nasza praca była tylko częściowo finansowana przez państwo.
Gdy w 1994 roku zrobiliśmy nasz prototyp, byłem pewny, że mniej więcej w tym czasie wszyscy będą mieli coś podobnego. Bardzo się zdziwiliśmy, gdy się okazało, że nikt nie tylko nie miał, ale nawet nie pomyślał, żeby to zrobić. Trochę się nawet baliśmy, że idea jest nietrafiona. Dopiero później zauważyliśmy, że dlatego „wybiegliśmy przed orkiestrę", gdyż inni, odnosząc sukcesy rynkowe, nie musieli myśleć o rozwoju swoich produktów. Wprawdzie już się zorientowali, ale dziś mamy około pieciu lat przewagi.

L.W.: *Na czym polega nowatorskość waszego rozwiązania?*

D.K.: Klasycznie zakłada się, że planowanie logistyczne dla przedsiębiorstw produkcyjnych należy realizować podobnie jak księgowość, a mianowicie centralnie. Jedno przedsiębiorstwo ma centralną gospodarkę finansami, jedną produkcję, a więc i jedną logistykę dla wszystkich procesów. Niezadowolenie producentów, które zauważyliśmy, wynika z faktu, że centralne planowanie logistyki jest mało praktyczne dla produkcji, a potrzeby wydziałów są zawsze inne niż planowano.

L.W.: *Czyli postanowiliście oprogramować logistykę z podziałem na określone procesy?*

D.K.: Tak. Kanonem jest centralna księgowość. Logistykę natomiast trzeba zdecentralizować dzieląc na realne procesy produkcji oraz pozwolić każdemu wydziałowi precyzować swoje logistyczne potrzeby. Natychmiast powstaje problem, jak zaprojektować procesy logistyczne, aby nie było problemów ze współpracą między wydziałami. W wyniku tych założeń powstało rozwiązanie, które nie należy do klasy MRP, a jest tylko szeregiem „małych" systemów MRP.

L.W.: *Jak to rozwiązanie ma się do SAP R/3?*

D.K.: Tu muszę wrócić do źródeł. Otóż nasze rozwiązanie miało trzech ojców. Byliśmy sami i razem z fachowcami ze wspomnianego instytutu, wszystko wymyśliliśmy i rozwinęliśmy. W 1993 r. bardzo interesował się naszym rozwiązaniem Siemens, który nam pomógł. Nie mniej ważne było to, że zwrócił na nas uwagę SAP, który sfinansował integrację naszego programu ILoS z SAP R/3.

To bardzo dobry początek dla małej spółki, gdy wspierają ją takie firmy jak SAP i Siemens.
Potem gdy zaczęliśmy wdrażać nasze rozwiązanie i klienci byli bardzo zadowoleni, SAP wycofał się z pomocy. Powiedziano nam, że już zbyt urośliśmy, aby nam pomagać. Jest szansa, że teraz znowu się nami zainteresują, ale na razie zastanawiają się, co z tym fantem zrobić.
Mamy umowy z firmami SAP i Siemens na dostarczanie naszego oprogramowania jako składnika ich projektów. Może być ono niekiedy użyte z SAP R/3 zamiast modułu PP lub może go uzupełniać. Ponad 50% naszych klientów ma IloSa, a nie ma R/3. Oferujemy nasz system niezależnie od tego, co ktoś aktualnie posiada. W tej chwili pracujemy nad uzupełnieniem w postaci własnego systemu księgowego.

L.W.: *Zaczęło się jednak skromnie. Kiedy, w ile osób i z czym zaczynaliście?*

D.K.: LIPRO stworzyliśmy w 1994 r. we czwórkę, mając w sumie 50 tys. DEM. Mieliśmy szczęście ponieważ trafiliśmy razem z Siemensem, - naszym partnerem - na pierwszych trzech dużych klientów. Dla dużych klientów robi się duże projekty i w rezultacie już w końcu pierwszego roku mieliśmy zysk wystarczający na nowe inwestycje. Powiększyliśmy kapitał najpierw do 100 tys., a w zeszłym roku już do 2 mln. DEM, a nasze przychody wyniosą w tym roku około 9 mln DEM. W tej chwili prowadzimy rozmowy z kolejnym inwestorem. Zyski należy inwestować - to nasza zasada.

L.W.: *W Polsce bardzo trudno zdobyć kredyty na rozwijanie oprogramowania, gdyż uważa się, że są to przedsięwzięcia zbyt wysokiego ryzyka. Jak zdobywa się kapitał w Niemczech na takie przedsięwzięcia?*

D.K.: Wiem z własnego, gorzkiego, doświadczenia, że w Niemczech jest pod tym względem tak samo jak w Polsce. Wprawdzie wszyscy mówią, że trzeba mieć zaufanie itd., ale rzeczywistość nie jest zbyt różowa. Na początku nikt nie dał nam ani grosza. Najważniejsi byli klienci, którzy z nami realizowali pierwsze projekty i dzięki temu mogliśmy inwestować w firmę. Duże znaczenie miało też to, że my wierzyliśmy, że się uda. Dopiero po dwóch kolejnych latach, gdy okazało się, że robimy dobrą robotę i przynosimy dochód, jeden z banków zdecydował się nam trochę pomóc.

L.W.: *Wiele u nas firm ma już jakiś system księgowy. Czy wasz IloS będzie mógł współpracować z innymi systemami f-k?*

D.K.: Mamy intrface do R/3 i kilku innych systemów f-k. Teraz opracowujemy interface do systemu f-k naszego partnera w Polsce - Budimex Soft S.A.

L.W.: *Macie więc dla IloSa polską wersję językową?*

D.K.: Tak. Bardzo szybko zauważyliśmy, że jak się robi pierwsze wdrożenia, to przyjdzie następny klient, na przykład wielki koncern z pytaniem o różne wersje językowe. Produkcja nie jest przecież dzisiaj narodowa tylko międzynarodowa. Zrobiliśmy już wersje: angielską, francuską, rosyjską, polską, a czeską właśnie kończymy.

L.W.: *W tworzeniu polskiej wersji mógł Pan również brać udział. Skąd Pan zna język polski?*

D.K.: W latach 70., kiedy byłem jeszcze studentem, przyjechałem do Polski i znalazłem tu wielu przyjaciół. Bardzo mi się w Polsce podobało i jako student często tu bywałem. Dlatego zacząłem się uczyć polskiego.

L.W.: *Czy dlatego postanowił Pan otworzyć tutaj przedstawicielstwo?*

D.K.: Moim zdaniem nasze oprogramowanie jest tutaj potrzebne i to jest najważniejsze. Na tę decyzję oczywiście wpłynęła także w pewnym stopniu chęć powrotu do młodości.

L.W.: *W jaki sposób chce Pan działać, czy poprzez pozyskiwanie partnerów wśród firm już istniejących, jak Budimex, czy przez stworzenie własnej silnej placówki?*

D.K.: Sukces można osiągnąć, pozyskując dobrych partnerów. Jeżeli jednak poważnie chcemy działać w Polsce, to trzeba także dużo robić samemu. Chcemy dość szybko przejść od przedstawicielstwa do samodzielnej spółki, nie tracąc naszego partnera.

L.W.: Przychylność polskiej społeczności, władzy i mediów zyskuje zagraniczny przedsiębiorca, gdy inwestuje w naszym kraju, a nie tylko sprzedaje swój produkt. Czy planuje Pan wykorzystanie przez LIPRO polskich inżynierów i techników?

D.K.: Oczywiście. My nie tylko sprzedajemy software, chcemy też być doradcą w szukaniu najlepszych rozwiązań organizacyjnych w przedsiębiorstwach. To jest nawet ważniejsze niż oprogramowanie. Jeżeli zarząd przedsiębiorstwa chce poprawić organizację, to najlepiej zrobić to, korzystając z pomocy zewnętrznej. Myślimy o wdrażaniu naszego oprogramowania, ale doradca musi być już z Polski. Nie wszystkie rozwiązania organizacyjne, które znamy i które sprawdzają się w Niemczech, mogą być metodą 1:1 przeniesione tutaj.

L.W.: *Jedną z cech IloSa jest możliwość parametryzacji. Kto będzie to robił?*

D.K.: Jest dopasowywany do potrzeb klienta. To jest bardzo ważne, ponieważ logistyka nie poddaje się standaryzacji. Warsztat w jednym zakładzie i w drugim to mogą być dwie całkiem różne rzeczy. Najpierw określamy razem z klientem jego potrzeby, zapisujemy to w formie dokumentacji i potem generujemy rozwiązanie potrzebne klientowi. Do wdrożeń oprócz partnerów komercyjnych potrzebujemy też naukowców i teoretyków. Mamy już pierwsze kontakty z Politechniką Warszawską i chcemy je rozwijać.

L.W.: *Jak układa się w Niemczech współpraca wyższych uczelni z firmami?*

D.K.: W Niemczech jest dosyć dobrze organizowana współpraca naukowców z firmami. Naukowcy próbują dopasować się do nowych warunków i potrafią pomóc przedsiębiorstwom w rozwijaniu nowych produktów. Wspólne przedsięwzięcia w dzie-

dzinie produkcji oprogramowania są częściowo finansowane przez państwo, a częściowo przez zainteresowane przedsiębiorstwa. W ostatnich latach w Niemczech trochę się to zmieniło, gdyż dominuje SAP ze swoim produktem R/3. Duży, ale jednak tylko jeden, producent nie wykorzysta wszystkich programistów. Ci pozostali, żeby przetrwać i przebić się z nowymi ideami, próbują znaleźć nowych odbiorców. Jest to bardzo ciekawa sytuacja, zwłaszcza dla mojej, względnie młodej, firmy.

L.W.: *Jak współpracuje LIPRO z instytucjami naukowymi?*

D.K.: W wyniku współpracy z Katedrą i Instytutem Badania Stosunków Pracy w Reńsko-Westfalskiej Wyższej Szkole Technicznej (RWTH) w Aachen, powstało w firmie Otto Fuchs studium dotyczące segmentyzacji produkcji i wprowadzenia systemu pracy grupowej. Planujemy kolejne projekty realizowane z instytucjami naukowymi. Obecnie w naszej firmie odbywają praktyki studenci z Berlińskiej Politechniki (TU) oraz Wyższej Szkoły Techniczno – Ekonomicznej w Berlinie (FHTW).

L.W.: *Pana firma uczestniczyła w międzynarodowym programie badawczym. Na czym polegał ten projekt i wasz w nim udział?*

D.K.: Tego typu projekty mają tworzyć platformę współpracy dla fachowców z różnych krajów Unii Europejskiej. Zostaliśmy zaproszeni do jednego z nich w 1996 r., kiedy byliśmy jeszcze małą firmą. Spostrzeżono bowiem, że nie istnieje wspomaganie komputerowe planowania budowy fabryk i zwykle robi się to „ręcznie". Projekt ten ma na celu stworzenie narzędzia typu CAD do projektowania fabryk. Model fabryki określa się do przewidywanej produkcji, na podstawie bazy danych o maszynach, urządzeniach i materiałach do produkcji. Nasz prototyp oprogramowania wykorzystano przy przebudowie szkockiej fabryki sprzętu gospodarstwa domowego Candy. Odbiorcy są bardzo zadowoleni.
Trzeba jeszcze jednak parę lat, aby powstał produkt komercyjny do tych zastosowań. Widać jednak już teraz, że będzie on wymagał dopasowania do każdego odbiorcy, jak też branży.

L.W.: *Jeżeli to nie jest tajemnica handlowa, to proszę powiedzieć, czy to zagadnienie znajduje się w kręgu waszych zainteresowań komercyjnych?*

D.K.: Może być Pan pewny, że oprogramowanie do projektowania fabryk będzie w naszej ofercie.

L.W.: *Przemysł software'owy ma taką specyfikę, że tylko jedna z dziesięciu firm osiąga sukces, ale za to bardzo duży. Często zdarza się, że zaczynając w przysłowiowym garażu, można stworzyć wielką firmę. Jak paru entuzjastów, praktycznie bez pieniędzy, może osiągnąć sukces rynkowy? Co by Pan poradził innym?*

D.K.: Po pierwsze trzeba wierzyć w swoje umiejętności i swoje rozwiązania. Po drugie trzeba znaleźć klientów, a najważniejsze aby wiedzieć, czego klienci naprawdę potrzebują. Trzeba mieć też tyle entuzjazmu dla swojego przedsięwzięcia, aby chęć konsumowania zysków odłożyć na później. Najtrudniejsze zaś jest znalezienie na początku partnerów - silnych, życzliwych partnerów. A potem wystarczy tylko z otwartymi oczami iść przez świat i szukać nowych możliwości.

L.W.: *Czy uważa Pan, że nadal, mimo istnienia gigantów przemysłu oprogramowania, sukces w tej branży „od początku" jest możliwy?*

D.K.: Jestem o tym przekonany, ponieważ rozwój i produkcja oprogramowania, a nawet cała informatyka, są dopiero na początku drogi swojego rozwoju. To się tylko może wydawać, że szanse sukcesu się wyczerpały. Powstają nowe kierunki i potrzeby, a giganci czasem szybko reagują na nie, a też bywa, że przesypiają odpowiedni moment. To trzeba i opłaca się wykorzystać.

L.W.: Na świecie jest duże zapotrzebowanie na produkty informatyczne, lecz nie nadąża z nim podaż programistów. Jaka jest sytuacja w Niemczech?

D.K.: W Niemczech powoli liczba dobrych programistów zaczyna być wąskim gardłem rozwoju. Jest to spowodowane tym, że zbyt wielu dobrych specjalistów zajmuje się pielęgnowaniem starych rozwiązań, a zbyt mało jest tych, którzy myślą o nowych rozwiązaniach.

L.W.: *Są małe, średnie firmy software'owe, giganci i tzw. firmy adwokackie, które tylko szukają kogo kupić, by na tym zarobić. Jak Pan widzi swoją firmę w kontekście częstych ostatnio fuzji i sprzedaży? Czy chciałby Pan, aby LIPRO stała się częścią dużego koncernu?*

D.K.: Są chętni, żeby kupić nasz produkt lub wykupić firmę, ale ani jedno, ani drugie nam nie odpowiada. Na przykład Siemens chciał przejąć większościowe udziały w LIPRO. Dwa lata temu nawet taką ewentualność rozważaliśmy, ale w końcu nie zgodziliśmy się i nadal nie zmieniamy zdania. Polityka wielkich firm software'owych nie zawsze orientuje się na nowe rozwiązania. Dość często kierują się tym, żeby bronić określony segment rynku czy zawierać strategiczne umowy z innymi. My natomiast chcemy poszukiwać i tworzyć najlepsze rozwiązania. Jesteśmy też przekonani, że znajdziemy takich partnerów finansowych, którzy będą wspierać realizację naszych własnych pomysłów.

L.W.: *Czyli wchłonięcie przez duży koncern małej firmy jest dla niej korzystne?*

D.K.: Nie ma na to reguły. Znam firmy, które stały się częścią dużego koncernu i to im pomogło, ale jest wiele takich, które na tym dużo straciły. My znaleźliśmy własną drogę i chcemy nią dalej podążać.

Dr Dieter Küchler (l. 49) ukończył Politechnikę Drezdeńską na wydziale Maszyny i Urządzenia Elektryczne. W roku 1974 uzyskał stopień doktora nauk technicznych. Temat pracy: „Przemysłowe techniki sterowania". Pracował naukowo tej dziedzinie w Berlińskim Instytucie Wysokich Mocy (IEA).
Szef ds. Rozwoju w Instytucie Elektrotechniki w Berlinie i szef projektów doradczych dla Siemensa, ABB i Elektro-Anlagen. Od 1990 prezes zarządu i udziałowiec spółki GESAT, należącej do znanej w Niemczech firmy consultingowej SYSKOPLAN. W tym czasie szef projektu prywatyzacji zakładów ELPRO AG.
Od 1992 prokurent w SYSKOPLAN w centrali firmy w Gutersloh. Odpowiedzialny za usługi consultingowe i rozwój systemu oprogramowania komputerowego ILoS dla sterowania produkcją. Od 1994 Prezes Zarządu i główny udziałowiec LIPRO GmbH w Berlinie.

Problemy z utrzymaniem dużych baz danych

Wojciech Czujowski, Jacek Gruber

Kontynuując rozważania o administrowaniu dużymi systemami bazy danych z artykułu pt. *Administrowanie dużymi bazami danych* („Informatyka" nr 9/98), koncentrujemy się obecnie na problemach administrowania dużymi systemami bazy danych (SBD) związanych fragmentacją i replikacją oraz alokacją replik i fragmentów dużych baz danych.

Mimo że administrowanie bazą danych ma miejsce w fazach eksploatacji i utrzymania cyklu życia systemu SBD, administrator bazy danych (ABD) ma dość duże możliwości optymalizowania wydajności, niezawodności i bezpieczeństwa systemu, o ile dysponuje tworzonymi systemowo statystykami z bieżącej eksploatacji dużego, rozproszonego systemu bazodanowego. Dlatego dokonamy też zwięzłego przeglądu narzędzi administracyjnych w wybranych systemach zarządzania bazami danych (SZBD), zwracając uwagę, czy istnieją tam możliwości zbierania takich statystyk.

Omówimy również kilka innych problemów administrowania. Zalecenia praktyczne dla administratora będziemy w tekście zaznaczać i starać się ich nie katalogować.

Po co i jak administrować rozproszeniem bazy?

Jednym z głównych celów administrowania dużą bazą danych jest *maksymalizacja wydajności całego systemu SBD*. W praktyce ABD może to realizować *na podstawie obserwacji i wniosków z eksploatacji i utrzymywania systemu SBD*. Administrator może również *wpływać na pielęgnację systemu, prowadząc bieżącą konserwację oraz pielęgnację danych bazy i dbając o jej bezpieczeństwo i niezawodność*.

W celu maksymalizacji wydajności i niezawodności SBD stosuje się alokację danych. *Na podstawie monitorowania statystyk wykorzystania bazy danych ABD może zmieniać alokację danych* środkami systemu SZBD. Niezbędne jest tu pewne minimum wiedzy ABD z zakresu algorytmów projektowych związanych z fragmentacją, replikacją i alokacją danych bazy. Należy podkreślić, że stosowanie dobrej alokacji wymaga również od ABD dużego doświadczenia zarówno w projektowaniu systemów SBD, jak i ich eksploatacji.

Alokacja

Projektowanie dużych SBD w znacznym stopniu polega na opracowaniu odpowiedniej alokacji danych oraz właściwym oprogramowaniu miejsc przetwarzania danych. Istnieją różne możliwości alokacji danych. Baza danych może być zreplikowana całkowicie lub częściowo albo utrzymywana jako pojedyncza kopia rozłożona na fragmenty.

Jak w każdym procesie projektowym inżynierii oprogramowania mamy tu dwie strategie projektowania: z góry na dół (*top - down*) i z dołu do góry (*bottom - up*). Strategię z góry do dołu stosuje się do projektowania SBD od podstaw, najczęściej dla heterogenicznego SZBD. Gdy integrujemy istniejące już bazy danych zarządzane przez różne SZBD w jeden logicznie spójny SBD, do projektowania stosujemy strategię z dołu do góry.

Zmiana alokacji istniejącej bazy danych następuje w wyniku analizy wykorzystania danych na podstawie statystyk lub miar wykorzystania danych z poszczególnych relacji. Często *statystyki te można zebrać za pomocą odpowiednich narzędzi systemowych administratora SZBD. Miary wykorzystania danych z poszczególnych relacji wyznacza się na podstawie wiedzy o SBD*.

Fragmentacja

Fragmentacja jest to podział fizyczny relacji na części rozproszone sieciowo. Struktura logiczna relacji pozostaje niezmieniona. Fragmentacja może zwiększyć efektywność przetwarzania danych. Rozróżniamy fragmentację poziomą (*horizontal*), pionową (*vertical*) i hybrydową (*hybrid*) za pomocą operacji algebraicznych selekcji i projekcji na podlegających fragmentacji relacjach.

Fragmentację stosuje się w celu zwiększenia dostępności danych. Zwiększenie dostępności polega na maksymalizacji lokalności odwołań do danych. Funkcja kosztu jest wtedy minimalizowana ze względu na komunikację między miejscami składowania danych. Ponieważ wykorzystanie danych przez poszczególne miejsca SBD może się zmieniać, ABD powinien obserwować statystyki wykorzystania fragmentów i poszczególnych atrybutów. Wymaga się, aby fragmentacja zachowywała reguły kompletności, rekonstrukcji i rozłączalności [1].

Fragmentacja pozioma jest podziałem instancji relacji na podzbiory krotek o tych samych atrybutach. Wskazane jest, aby realizację tego podziału na podzbiory wykorzystywanych krotek zmieniać, ponieważ podczas eksploatacji zmieniają się wymagania informacyjne SBD. *W celu zmiany sposobu fragmentacji poziomej ABD powinien analizować statystki komunikacji dla operacji selekcji.*

Fragmentacja pozioma może być fragmentacją pierwotną (*primary horizontal fragmentation*) i wtórną (*derived horizontal fragmentation*).

Aby wyznaczyć fragmentację poziomą, powinniśmy znać liczność card(R) relacji R oraz posiadać informację kwalifikującą i ilościową. Informacją kwalifikującą jest zapytanie zdefiniowane na relacji R. Informacja ilościowa to dwa zbiory danych: selektywność oraz częstość dostępu minitermu do krotek relacji. Selektywność sel(m_i) minitermu m_i określa liczbę krotek w relacji, które byłyby dostępne poprzez zapytanie wyspecyfikowane na podstawie predykatu minitermowego. Częstość dostępu acc(q_i) zapytania q_i określa średni czas pomiędzy żądaniami dostępu do krotek relacji.

Pierwotna fragmentacja pozioma wynika z zastosowania predykatu minitermowego zdefiniowanego na atrybutach relacji. Zatem podział relacji na fragmenty zależy od definicji predykatów minitermowych. Zbiór fragmentów poziomych spełniających te predykaty nazywamy zbiorem fragmentów minitermowych. Zbiór predykatów prostych jest kompletny, jeżeli umożliwia to każdej aplikacji dostęp - z jednakowym prawdopodobieństwem - do jakichkolwiek dwóch krotek należących do jakiegokolwiek fragmentu minitermowego. Zbiór predykatów prostych powinien być minimalny. Do generowania kompletnych minimalnych zbiorów predykatów prostych stosuje się algorytm COM_MIN. Dokonuje on podziału relacji i wyszukuje predykaty, tak aby otrzymać minimalny i kompletny ich zbiór. W drugim kroku algorytm produkuje zbiór predykatów minitermowych, które następnie przechodzą eliminację ze względu na znaczenie. Do wyznaczania fragmentacji pierwotnej stosuje się algorytm PHORIZONTAL, który wybiera odpowiednie predykaty minitermowe. Zbiór tak wybranych predykatów dzieli relację na odpowiednią liczbę fragmentów.

Wtórna fragmentacja pozioma jest podziałem danej relacji, spowodowanym zastosowaniem predykatu zdefiniowanego na innej relacji. *Aby dokonać wtórnej fragmentacji poziomej, ABD musi znać sposób podziału relacji nadrzędnej oraz podrzędnej.* Fragmentacja jest oczywista w wypadku, gdy każda krotka z relacji podrzędnej ma jedno powiązanie do relacji nadrzędnej. Jeżeli krotka podrzędnej relacji ma dwa lub więcej powiązań do relacji nadrzędnej, to wybór fragmentacji spośród fragmentacji kandydujących opiera się na charakterze i statystykach wykorzystania powiązania. Złączenia relacji nadrzędnej i podrzędnej są definiowane jako równozłączenia (*equi - join*), które mogą być realizowane jako półzłączenie (*semijoin*).

Fragmentacja pozioma jest poprawna, jeżeli spełnia wspomniane wcześniej kryteria. Zachowana jest przy tym reguła rekonstrukcji. Rekonstrukcji można dokonać przez zastosowanie operatora unii. Kompletność opiera się na zbiorze predykatów selekcji, który z założenia jest kompletny i minimalny. Reguła rozłączalności jest zachowana, jeżeli predykaty minitermowe wykluczają się.

Fragmentacja pionowa dzieli relację R na fragmenty R_1, R_2,... R_n, które zawierają podzbiory atrybutów relacji R. Podziału dokonujemy tak, aby każda z aplikacji działała tylko na jednym fragmencie R_i. Podział pionowy jest bardziej złożony. Dlatego rozwiązanie problemu fragmentacji pionowej opiera się częściowo na heurystykach: grupowaniu (*clustering*) i rozdzielaniu (*spliting*). Grupowanie polega na rozdzieleniu atrybutów i łączeniu ich według założonych kryteriów. O rozdzielaniu atrybutów decyduje się na podstawie założeń o dostępie do grupy atrybutów z konkretnej aplikacji.

Aby wyznaczyć fragmentację pionową ABD musi poznać aplikacje SBD. W szczególności należy grupować atrybuty wykorzystywane przez poszczególne aplikacje. Miarą powiązań wynikających z częstości użycia atrybutów przez aplikacje jest afiniczność (*affinity*).

Do rozwiązania problemu fragmentacji pionowej stosuje się algorytm grupowania BEA (*bond energy algorithm*) [2]. Algorytm BEA grupuje atrybuty o małej i dużej afiniczności, dokonując przekształceń na macierzy afiniczności AA do postaci sklastrowanej macierzy afiniczności CA (*clustered affinity measure*). Kolejno są permutowane kolumny, a następnie wiersze, aż do osiągnięcia maksymalnej wartości globalnej miary afiniczności. Algorytm produkuje sklastrowaną macierz afiniczności, w której wyraźnie pogrupowane są atrybuty o dużej i małej afiniczności.

Sklastrowana macierz afiniczności musi być algorytmicznie podzielona na zbiory atrybutów, które są dostępne przez poszczególne aplikacje. Zbiór atrybutów może się zmieniać w czasie użytkowania SBD. Dlatego *ABD powinien obserwować statystyki związane z operacjami projekcji.* Do wyznaczania punktu rozszczepienia sklastrowanej macierzy afiniczności stosuje się algorytm podziału PARTITION.

Dąży on do maksymalizacji dostępu do atrybutów we fragmencie i minimalizacji całkowitego dostępu do fragmentów na podstawie całkowitej liczby dostępów do atrybutów przez aplikację.

W wypadku fragmentacji pionowej występuje dodatkowo problem atrybutów łączących fragmenty pionowe w celu zachowania reguły rekonstrukcji. W tym celu replikuje się atrybuty kluczowe albo można użyć identyfikatorów nadawanych przez SZBD. Algorytm PARTITION gwarantuje też inne cechy poprawności fragmentacji: kompletność i rozłączalność. Rozłączalność jest określana bez uwzględniania atrybutów kluczowych.

Fragmentacja hybrydowa łączy zasady fragmentacji pionowej i poziomej. Można tu zastosować kolejno fragmentację poziomą i pionową. Algorytmy i reguły wyznaczania fragmentów są takie same, jak dla fragmentacji poziomej i pionowej. *Należy jednak pamiętać, że zbyt duża liczba małych fragmentów może powodować bardzo duże koszty złączeń.* Ponieważ fragmentacji hybrydowej dokonuje się za pomocą tych samych, odpowiednio połączonych algorytmów fragmentacji poziomej i pionowej, to poprawność fragmentacji można wykazać na podstawie tych samych założeń i własności.

Replikacja

Replikacja jest to zwielokrotnienie fizyczne umiejscowienia tej samej informacji w różnych miejscach sieciowych. Replikacja zwiększa dostępność i niezawodność systemu bazodanowego. Replikowana może być cała baza danych lub poszczególne rela-

cje. Z replikacją wiąże się zagadnienie utrzymania zgodności replik, które opiera się na propagacji zmian (*update propagation*).

Replikację stosuje się w celu maksymalizacji optymalności działania. Za miary tej optymalności przyjmuje się: funkcję kosztu oraz wydajność. Na funkcję kosztu składają się: koszt przechowywania fragmentu w danych miejscu, koszt aktualizacji we wszystkich miejscach pamiętania replik w celu utrzymania integralności (*intgrity enforcement cost*) oraz koszt dostępu (*access cost*). Replikacja jest uzasadniona, jeśli koszt transmisji i aktualizacji danych jest większy niż koszt umieszczenia repliki w kolejnym miejscu przetwarzania. Utworzenie repliki maksymalizuje dostępność do bazy danych i wydajność SBD. *Problem alokacji fragmentów bazy danych NP-trudny i przy jego rozwiązywaniu korzysta się z heurystyk.* Heurystyki zalecają między innymi traktowanie fragmentów jako osobnych plików. Należy jednak pamiętać o powiązaniach między fragmentami. Zaleca się zapewnianie ścisłego powiązania pomiędzy miejscami alokowania fragmentów a miejscami przetwarzaniem zapytań, co oznacza, że żądania dostępu do danych powinny być przetwarzane w miejscu przechowywania fragmentu. Posiadając wiedzę o SBD, administrator może w ten sposób minimalizować koszty transmisji danych pomiędzy miejscami. Nie ma formalnych modeli i algorytmów mogących, które na podstawie wiedzy o fragmentach mogłyby generować jednocześnie optymalne sposoby alokacji i replikacji.

Do utworzenia replik niezbędne są informacje o bazie danych oraz aplikacjach SBD. Administrator musi znać relacje wykorzystywane przez poszczególne aplikacje oraz ich powiązania. Niezbędne są również informacje statystyczne dotyczące bazy danych, na przykład ilość dostępów wykonywanych z aplikacji. Podobnie jak w wypadku alokacji trzeba mieć charakterystykę miejsca, w którym znajdzie się replika, tj. wielkość pamięci, parametry przetwarzania, koszt przechowywania i transmisji danych.

Replikę całej bazy danych można tworzyć w celu zwiększenia niezawodności systemu bazodanowego, dostępności bazy danych oraz w celu zwiększenia wydajności SBD. Ważną miarą jest również koszt utrzymywania spójności danych.

W wypadku dużych baz danych często stosuje się tworzenie repliki całej bazy danych (*database mirroring*) w specjalnie przygotowanym miejscu, które przejmuje w razie awarii rolę serwera macierzystego bazy danych (*primary servwer*). Baza danych w takim miejscu jest na bieżąco uaktualniana i wykorzystywana do odczytu lub tylko awaryjnie. Zwiększa to znacznie niezawodność SBD.

Uwagi i zalecenia dla administratora dużej bazy danych dotyczące zmiany alokacji:

- Jeśli alokacja ma być optymalna, ABD powinien monitorować i analizować za pomocą dostępnych narzędzi SZBD statystyki wykorzystania istniejących już fragmentów oraz replik;
- Fragmenty i repliki bazy danych należy umieszczać w miejscach o maksymalnej dostępności i niezawodności;
- Zbyt duża liczba fragmentów może powodować duże koszty złączeń, czyli zmniejszenie dostępności;
- Replikacja całej bazy w miejscach przetwarzania zwiększa dostępność do bazy danych i wydajność przetwarzania SBD.

W praktyce, w celu uproszczenia problemu alokacji, stosuje się najpierw optymalizację bez uwzględnienia replikacji. Replikację rozpatruje się w następnym kroku, w którym maksymalizuje się miary niezawodności bazy danych.

Dla dużych baz danych stosuje się replikację całej bazy danych w celu maksymalizacji niezawodności.

Bezpieczeństwo

Problem bezpieczeństwa bazy danych należy rozpatrywać w dwóch aspektach: ochrony danych przez nieautoryzowanym dostępem oraz ochrony przed naruszeniem reguł integralności danych.

Ochrona danych przed niepowołanym dostępem jest realizowana na poziomie systemu operacyjnego oraz SZBD i SBD. Stosuje się w tym celu mechanizmy autoryzacji użytkowników, ustalania praw przyłączenia się ich do bazy, określenia wielkości dostępnej przestrzeni dyskowej dla wykorzystywanych obiektów, specyfikację dozwolonych dla użytkownika operacji systemowych, określenie limitu wykorzystywanych zasobów.

Zabezpieczenia realizowane są na dwóch poziomach - na poziomie użytkownika (*user - level security*) - gdy użytkownikom są przydzielone hasła - *na poziomie dostępu dzielonego* (*share - level security*) - w wypadku haseł związanych z prawami do obiektów SZBD i prawami do operacji na tych obiektach.

Do ochrony przed nieautoryzowanym dostępem ABD może wykorzystać następujące, zwykle istniejące w SZBD, zabezpieczenia: ochronę przed nieautoryzowanym dostępem do bazy, ochronę przed nieautoryzowanym dostępem do obiektów schematu bazy, kontrolę wykorzystywania dysku, kontrolę używania zasobów systemu, śledzenie obiektów bazy danych.

Drugi aspekt to *zabezpieczenia przed naruszeniem przez użytkownika reguł integralności danych.* ABD powinien określić, którzy użytkownicy mogą mieć dostęp do schematu bazy, rodzaj udostępnianych akcji. ABD powinien również obserwować na bieżąco obiekty schematu bazy.

Przed naruszeniami reguł integralności bazy danych przez użytkowników zabezpieczają definiowane na poziomie SZBD reguły integralności semantycznej oraz mechanizmy wymuszanie więzów integralności. W dużych, rozproszonych bazach danych jest to zagadnienie szczególnie skomplikowane. *Reguły integralności mogą być wymuszane statycznie i dynamicznie.* Statyczna kontrola integralności semantycznej wynika z własności relacji. Odrzucane są akcje na bazie danych powodujące niespójność. Dynamiczne wymuszanie reguł integralności uruchamia akcje, które przywracają spójność bazy danych. Można w ten sposób wykrywać niespójność lub zapobiegać niespójności. W istniejących SZBD stosuje się deklarowane więzy integralności (*rule check*) oraz mechanizmy wymuszania integralności przez wyzwalacze (*trigger*).

Uwagi i zalecenia dla ABD dużej bazy danych dotyczące zabezpieczeń:

- Zaleca się ustalanie minimalnych niezbędnych praw dostępu do obiektów bazy danych dla poszczególnych użytkowników;
- Określając uprawnienia dostępu do bazy danych, należy stosować hierarchię użytkowników oraz ograniczyć liczbę tych, którzy mogą kreować nowych użytkowników, nadawać uprawnienia i zarządzać systemami SZBD i SBD;

- W celu ochrony szczególnie ważnych danych trzeba stosować zabezpieczanie obiektów bazy danych;
- Należy zabezpieczyć dostęp do danych z poziomu systemu operacyjnego.

Archiwizacja i przywracanie dostępności danych

Jednym z podstawowych zadań administratora jest zapewnienie dostępności danych. Dostępność jest obniżana między innymi przez występujące uszkodzenia sprzętowe, defekty i uszkodzenia w oprogramowaniu, błędy ludzi. Uszkodzenie bazy danych doprowadza często do konieczności odtworzenia z kopii archiwalnej. W celu uniknięcia archiwizacji i odtwarzania dużych baz danych stosuje się ich kopie „lustrzane", ale po uszkodzeniu tzw. bazy głównej należy ją jednak również odtworzyć.

Odtwarzanie danych po awarii i przywrócenie dostępności bazy danych powinno być realizowane w następujących krokach:

- stwierdzenie, czy: baza danych zawiera błędne dane, jest uszkodzona, dane zostały utracone;
- zatrzymanie wszystkich niezbędnych transakcji;
- stwierdzenie źródła i zasięgu uszkodzeń;
- przywrócenie zasobów systemu do stanu sprzed awarii;
- naprawienie uszkodzonych danych lub usunięcie danych niepoprawnych;
- ponowne uruchomienie lub wznowienie przerwanych transakcji;
- doprowadzenie do powrotu do normalnej pracy systemu.

Stosuje się kilka technik przywracania dostępności bazy: archiwizacja całej bazy i ponowne uruchomienie SZBD (*dump and restart*), cofnij-przywróć (*undo - redo*), ponowne załadowanie i ponowne uruchomienie (*reload and re - execute*), odtworzenie i powtórzenie (*restore and repeat*) [3].

Większość SZBD umożliwia archiwizację i odtwarzanie z kopii archiwalnej. W wypadku dużych baz danych pojawiają się jednak wymagania dla ABD i narzędzi SZBD. W dużych systemach baz danych nie zawsze jest konieczne odtwarzanie całościowe. Archiwizowanie nie powinno też mieć znaczącego wpływu na działanie SZBD.

Uwagi i zalecenia dla ABD dużej bazy danych dotyczące przywracania dostępności:

- Należy opracować procedury postępowania dla wykonywania kopii archiwalnej oraz odtwarzania po awarii;
- Archiwizowanie i odtwarzanie z kopii archiwalnej dużej bazy danych jest bardzo czasochłonne.

Narzędzia administrowania w systemach SZBD

Administrator ma do wypełniania wiele różnorodnych zadań. Zaliczamy do nich: definiowanie oraz wprowadzanie standardowych działań i procedur postępowania dla użytkowników, weryfikację działania SBD i/lub SZBD, administrowanie i monitorowanie bazy danych [4]. *Do realizacji tych zadań administrator wykorzystuje narzędzia istniejące w SZBD. Ukonkretnienie zadań administrowania bazą danych wynika z wniosków dotyczących działania SBD i/lub SZBD.*

ABD zarządza wzajemnymi powiązaniami obiektów bazy danych: tabel, indeksów, widoków, zdarzeń, wyzwalaczy, procedur, zabezpieczeń i innych obiektów SZBD. *Praktyka wskazuje, że niezbędne jest w tym celu dysponowanie rozbudowanym narzędziem współpracującym z SZBD.* Oprogramowanie to powinno umożliwiać zarządzanie i kontrolowanie działanie serwera bazy danych. Zarządzanie całą bazą danych oraz różnymi jej obiektami powinno być realizowane za pomocą interfejsu graficznego (*graphical user interfeace - GUI*). Zasadą powinno być, że narzędzie administrowania serwerem bazy danych ukrywa syntaktyczne szczegóły komend SQL używanych do tworzenia, zmiany, usuwania baz danych, relacji, indeksów, ograniczeń, zdarzeń, zasad, wyzwalaczy, procedur, praw dostępu. Jednocześnie powinna istnieć możliwość używania języka SQL.

Do efektywnej, bieżącej kontroli i analizy działania serwera niezbędne jest również narzędzie do tworzenia statystyk SZBD. Statystyki można wtedy wykorzystywać w procesie tzw. strojenia bazy danych.

Charakterystyka narzędzi administrowania bazami danych w wybranych systemach SZBD

Poniżej dokonamy przeglądu narzędzi administracyjnych dostarczanych w sześciu wybranych i najczęściej (wg [5]) używanych serwerach baz danych: CA-OpenIngres, IBM DB2, Informix, Microsoft SQL Server, Oracle, Sybase SQL Server (tabela 1). Ponieważ minął już pewien czas od momentu przygotowania tego krótkiego przeglądu, niektóre szczegóły tej charakterystyki mogą się nieznacznie różnić od tych z najnowszych wersji tych serwerów baz danych.

CA-OpenIngres

Wersja 1.1 i 1.2 CA - OpenIngres zawiera graficzne narzędzie zarządzania obiektami bazy danych DBA - Visual. ABD ma również do dyspozycji następujące narzędzia dodatkowe: CA - Unicenter,

Zestawienie analizowanych SZBD

SZDB	Producent
CA – OpenIngres 1.2	Computer Associates International Inc. (Islandia, NY)
IBM DB2 2.1.1	IBM Corp. (Armonk, NY)
Informix-OnLine Workgroup Server	Informix Software Inc. (Menlo Park, CA)
Oracle 7 ver. 7.3	Oracle Corp. (Redwood Shores, CA)
SQL Server 6.5	Microsoft Corp. (Redmond,WA)
Sybase SQL Server 11	Sybase Inc. (Emeryville, CA)

Configuration-by-Forms, CA - OpenIngres Control Program, CA - OpenIngres/Replicator, CA-OpenIngres/Star i Interactive Performance Monitor. Dają one możliwość monitorowania i strojenia bazy danych oraz zarządzania procesami serwera bazy danych.

Kreator DBA-Visual służy do tworzenia, modyfikowania i usuwania baz danych, tabel, umiejscowienia, użytkowników, grup, ról, indeksów, powiązań, widoków, reguł (odpowiedniki wyzwalaczy), procedur, zdarzeń i praw dostępu. Podczas działania kreator wytwarza automatycznie kod w języku zapytań podobnym do standardu SQL. Interfejs użytkownika pozwala wykorzystywać zaawansowane techniki manipulowania wszystkimi graficznie reprezentowanymi obiektami bazy danych.

Narzędzie Configuration-by-Forms jest przeznaczone do ustalania parametrów serwera bazy danych, serwerów rozproszonych baz danych, procesów i serwerów archiwizacji oraz szczegółów dotyczących pliku logu transakcji.

CA-Unicenters jest zintegrowanym narzędziem zarządzania zawierającym:

- Security Manager - narzędzie administrowania użytkownikami, dostępem do bazy danych i plików, zabezpieczeniami na poziomie C2, prowadzenie dziennika dostępu na różnych poziomach;
- Network Management - służy do administrowania połączeniami za pomocą protokołów sieciowych, np. TCP/IP, SNA, DECnet;
- Intelligent Database Agents - inteligentni agenci pozwalają wykorzystywać zasoby baz danych w innych standardach SZBD, np. DB2, Oracle, Informix, Sybase;
- Event Management - śledzi i raportuje zdarzenia związane z SZBD, SBD i komunikacją sieciową. Do powiadamiania ABD może używać poczty elektronicznej;
- Storage Management - umożliwia pełną lub przyrostową archiwizację i odtwarzanie bazy danych z kopii archiwalnej;
- Workload Management - to kalendarz oraz plan zdarzeń;
- Change Management - zarządza i monitoruje dostępne zasoby oraz pozwala zarządzać użytkownikami w środowisku rozproszonym;
- Performance Management - za pomocą pewnych mechanizmów zarządzania pozwala rozwiązywać problemy związane z wydajnością SZBD;
- Output Management - raportuje na bieżąco o stanie bazy danych, widokach, zapytaniach. Informacje te mogą być przesyłane do ABD pocztą elektroniczną.

CA-OpenIngres/Replicator pozwala tworzyć przezroczyste dla aplikacji użytkownika repliki danych - lokalnie lub w innych miejscach sieci. Replikator współpracuje też z innymi SZBD, co pozwala budować grupy serwerów baz danych kontrolowanych za pomocą tego narzędzia. Replicator daje możliwość tworzenia trzech typów replikacji: równorzędne (*peer-to-peer*), główny/podrzędny (*master/slave*), centralny/oddział (*central/branch*). Do zarządzania rozproszoną bazą danych służy CA-OpenIngres/Star. Gdy następuje aktualizowanie tabel, uruchamia on optymalizator zapytań rozproszonych, używając przy tym protokołu 2PC.

Interactive Preformance Monitor zbiera statystyki różnorodnych wskaźników wpływających na działanie SZBD.

IBM DB2

DB 2 w wersji 2.1.1 zawiera Administrator Toolkit, który jest zbiorem narzędzi dla ABD. Mogą być one zainstalowane w miejscu, w którym przechowywana jest baza danych. Za pomocą Administrator Toolkit administrowanie można realizować zdalnie.

Administrator Toolkit zawiera narzędzie Database Director umożliwiające - za pomocą interfejsu graficznego - konfigurowanie, archiwizację i odtwarzanie bazy danych. Można wybrać jeden lub więcej obiektów baz danych, które następnie można przekazać do DBA Utility, Preformance Monitor lub Visual Explain. Database Director pokazuje obiekty bazy danych jako powiązane ze sobą tabele oraz umożliwia tworzenie i modyfikowanie baz danych, przestrzeni tabel i tabel. Wszystkie inne obiekty bazy danych muszą być tworzone przy użyciu DB2 Command Line Processor.

DBA Utility służy do konfigurowania parametrów bazy danych, np. wielkości pliku logu bazy danych, przestrzeni tabel. Preformance Monitor umożliwia mierzenie, prezentowanie i analizę około 200 cech i wskaźników związanych z obiektami bazy danych. Visual Explain to narzędzie graficzne prezentujące plan dostępu ustalony przez optymalizator zapytań.

Do obsługi replikacji IBM oferuje Distributed Database Connection Services. Utrzymywanie integralności bazy danych jest realizowane za pomocą zaimplementowanego protokołu 2PC. Dane w replikach są uzgadniane synchronicznie przez DataPropagator Relational, DataPropagator NonRelational oraz DataRefresher, które wchodzą w skład programu narzędziowego Data Replication. DataJoiner natomiast umożliwia połączenie z różnymi SZBD, tj. DB2, Informix, Sybase, Oracle, Microsoft SQL Server.

Informix OnLine

Informix-OnLine Workgroup Server zawiera Command Center, które jest zbiorem narzędzi służących do administrowania serwerem bazy danych. W jego skład wchodzą: Space Explorer, Setup, Database Explorer. Narzędzia zawarte w Command Center mogą być uruchamiane osobno.

Używając Command Center, otrzymujemy informacje o aktualnym stanie serwera bazy danych oraz listę aktywnych użytkowników i otwartych sesji. Space Explorer służy do tworzenia, modyfikowania i usuwania baz danych, przestrzeni tabel oraz tabel. Udostępnia on również informacje o replikach. Setup pozwala zmieniać hasła, ustalać protokoły transmisji, zarządzać użytkownikami.

Informix Relational Object Manager (ROM) jest zbiorem narzędzi umożliwiających tworzenie i zarządzanie bazami danych w środowisku graficznym. ROM składa się z: Database Explorer, Table Editor, SQL Editor. Database Explorer umożliwia tworzenie, edycję, usuwanie baz danych, tabel, procedur, widoków, wyzwalaczy. Table Editor stosujemy do edycji i tworzenia tych obiektów bazy danych, do których brak jest dostępu za pomocą Database Explorer, np. kolumn tabel oraz indeksów. Równoważne możliwości daje Informix Command Line Processor. Natomiast za pomocą języka zapytań można manipulować wszystkimi wymienionymi obiektami baz danych.

Microsoft SQL Server

Microsoft SQL Server posiada zintegrowane środowisko zarządzania bazami danych SQL Server Enterprise Manager. Poza tym mamy tu: SQL Setup, SQL Service Manager, SQL/w, SQL Security Manager, SQL Transfer Manager i SQL Preformance Monitor.

Za pomocą Enterprise Manager kolejne poziomy obiektów bazy danych są przedstawione w postaci drzewa. Na najwyższym poziomie znajdują się zarejestrowane SQL Serwery. Z każdym serwerem związane są niższe poziomy: urządzenie składowania bazy danych (*database device*), urządzenie składowania kopii zapasowej (*backup device*), bazy danych, użytkownicy. Kolejne poziomy składają się ze szczegółów. Dla bazy danych otrzymujemy listę tabel, widoków, procedur składowana danych, reguł, domyślnych i definiowanych przez użytkownika typów danych. Każdy obiekt posiada listę operacji, które można na nim wykonać. Na przykład operacje dla tabeli umożliwiają: utworzenie nowej tabeli, modyfikację, usunięcie, zmianę nazwy, utworzenie indeksu, wyzwalacza, ustalenie prawa, zależności, wygenerowanie skryptu w języku zapytań. Skrypt w języku zapytań zawiera wszystkie definicje, łącznie z więzami integralności, indeksami, prawami dostępu. Niektóre z opcji operacji dla poszczególnych obiektów bazy danych są również dostępne w inny sposób, tzn. bezpośrednio z poziomu narzędzia. Składowanie w kopii zapasowej może odbywać się na poziomie bazy danych lub tabel. Archiwizacja może być wykonywana automatycznie - w chwili gdy tabela zostanie zmieniona. ABD może monitorować działania SZBD, a odpowiednie informacje o zdarzeniach w SZBD otrzymywać za pomocą poczty elektronicznej.

SQL Setup używany do rekonfiguracji serwera, np. zmiany opcji sieciowych, odbudowania bazy master, zmiany zbioru znaków lub sposobu ich sortowania, ustawienia opcji serwera, ustawienia opcji zabezpieczeń, usunięcia SQL Servera.

SQL Service Manager jest przeznaczony do uruchamiania, wstrzymywania, wznawiania i zatrzymywania pracy SQL Server i SQL Executive.

Za pomocą SQL/w można wprowadzać wyrażenia w języku Transact - SQL oraz procedury formułowane za pomocą graficznego interfejsu zapytań.

SQL Security Manager pozwala zarządzać kontami użytkowników SQL Server. Możliwe jest zdefiniowanie kontroli bezpieczeństwa zintegrowanej z systemem Windows NT oraz oddzielnie - tylko dla serwera bazy danych.

SQL Transfer Manager dostarcza prostych graficznych narzędzi do przenoszenia obiektów i danych z jednego do innego serwera baz danych Microsoft SQL Server.

SQL Preformance Monitor łączy Windows NT Preformance Monitor z SQL Server oraz dostarcza dane o aktywności systemu i statystyki serwera.

Komunikacja z serwerem może odbywać się przez sterowniki ODBC. Zapewniona jest obsługa komunikacji poprzez TCP/IP Sockets, Named-Pipes, NWLink IPX/SPX, Apple Talk ADSP, Banyan VINES. Replikacja opiera się na protokołach publikowania i subskrybowania części lub całości tabel z fragmentacją poziomą lub pionową. Transakcje rozproszone są obsługiwane przez Microsoft Distributed Transaction Coordinator, który działa na podstawie protokołów 2PC.

Oracle

Wszystkie narzędzia zarządzania dostępne w SZBD Oracle 7 są narzędziami graficznymi. W systemie dostępne są:

- Database Manager - narzędzie do uruchamiania i zatrzymywania serwera bazy danych, zmiany modelu pamięci, ustawiania rozmiaru buforów, rozmiaru pamięci tymczasowej, parametrów podłączania do serwera;
- User Manager - służy do tworzenia użytkowników, określania ich ról, ustalania praw ;
- Object Manager - zarządza obiektami baz danych, tj. tabelami, indeksami, synonimami, widokami, prawami do obiektów;
- Session Manager - prezentuje i odłącza użytkowników bazy danych;
- Database Expander - rozszerza bazę danych dla dodatkowych danych;
- Backup and Recovery Managers - do archiwizacji i odtwarzania danych;
- Password Manager - zmienia hasła SZBD.

Wersja Oracle 7 jest rozszerzona o współpracę z protokołem pocztowym SNMP oraz o zintegrowane graficznie narzędzie Oracle Enterprise Manager, składające się z następujących części:

- Navigator - obiektowa przeglądarka pokazująca drzewo powiązań baz danych, użytkowników, praw, profilów;
- Map Window - służy do zarządzania i monitorowania kluczowych obiektów bazy danych, które mogą być grupowane według różnych kryteriów;
- Job Scheduling System - kalendarz uruchamiający automatycznie procedury zdarzeń, powtórzeń zadań;
- Event Management System - jest monitorem baz danych i zdarzeń systemowych.

Oracle7 Enterprise Backup Utility obsługuje archiwizowanie i odtwarzanie bazy danych po awarii. Oracle Transparent Gateways i Oracle Procedural Gateways integrują w pełni SZBD Oracle z DB2, Informix, CA-Ingres i Sybase. Oracle7 Advanced Replication Option pomaga zarządzać replikacjami, które opierają się na publikacji i subskrypcji danych. Umożliwia on dynamiczne śledzenie zmian w bazie danych w miejscach sieciowych.

Sybase SQL Server

Sybase SQL Server 11 oferuje kilka narzędzi ABD. Są to: Service Manager, Server Config Utility, Server Manager, ISQL.

Za pomocą Service Manager można zatrzymywać i uruchamiać serwer bazy danych, a także ustalać sposób monitorowania, ustalać połączenia sieciowe, wybierać serwer archiwizacji, inicjować log błędów, log zdarzeń i log bezpieczeństwa dostępu.

Server Manager służy do zarządzania obiektami bazy danych: tabelami, widokami, indeksami, wyzwalaczami, procedurami, regułami, typami definiowanymi przez użytkownika, a także użytkownikami i grupami.

Sybase posiada osobny serwer archiwizacji kopii zapasowej Sybase Backup Server. Jako osobny produkt oferowany jest również Sybase Enterprise SQL Server Manager. Jest to rozbudowane narzędzie zarządzania bazami danych z wieloma rzadko gdzie indziej spotykanymi funkcjami administracyjnymi.

DirectConnect jest podłączeniem Sybase do Adabas, DB2, CA-OpenIngres, Informix, Oracle, i VSAM. Sybase Replication Server używa tego narzędzia do asynchronicznej homo- i heterogenicznej replikacji danych. Możliwe jest monitorowanie informacji o replikacji.

Podsumowanie przeglądu

Wszystkie omówione SZBD oferują narzędzia z podstawowymi funkcjami zarządzania serwerem bazy danych. Zasadniczo wspomagają one zadania technicznego zarządzania bazami danych, tworzenia, modyfikowania i usuwania obiektów SZBD, tworzenia kopii zapasowych i odtwarzania.

Produkty CA - OpenIngers, Informix, Oracle, Microsoft SQL Server, Sybase pozwalają na administrowanie za pomocą narzędzi typu „explorer". Narzędzia te ułatwiają manipulowanie bazami danych oraz obiektami zawartymi w bazach.

CA - Visual DBA/CA - OpenIngres, Microsoft SQL Server Enterprise Manager, Sybase Enterprise SQL Server Manager mają możliwość zarządzania zadaniami SZBD.

Wszystkie omawiane tu SZBD umożliwiają dostęp i zarządzanie za pomocą języka zapytań zgodnego lub zbliżonego do standardu SQL. Do przetwarzania transakcji wykorzystuje się protokół 2PC.

Większość SZBD posiada rozbudowane moduły zbierania statystyk serwera bazy danych. Żaden SZBD nie oferuje narzędzia wspomagającego analizowanie alokacji danych.

Znacznie rozbudowane są narzędzia i mechanizmy fragmentacji. W niektórych wypadkach (np. Sybase) występują osobne serwery replikacji do zarządzania synchronizacją i monitorowaniem.

Niedostatki narzędzi administrowania

Z konstrukcji istniejących SZBD - z których część omówiliśmy wyżej - wynikają ich braki w zakresie zarządzania fragmentacją i replikacją baz danych. Jest to szczególnie kłopotliwe, gdy administruje się dużym rozproszonym systemem baz danych.

Wykonano zatem projekt modułu administratora dużego rozproszonego systemu baz danych. Założono, że ma istnieć możliwość zarządzania fizyczną alokacją bazy danych i w tym celu będzie się wykorzystywać statystyki oraz algorytmy, które omówiliśmy wcześniej. Wykonany projekt opiera się też na zaleceniach dla administratora, które staraliśmy się wyżej systematycznie formułować.

Główną wadą istniejących narzędzi ABD jest to, że nie wykorzystują zbieranych statystyk. Żaden przedstawionych wyżej SZBD nie oferuje narzędzia wspomagającego analizę i projektowanie alokacji danych na podstawie zebranych statystyk eksploatacyjnych.

W zaprojektowanym zintegrowanym narzędziu uwzględniono funkcje wspierające zadania administratora w zarządzaniu obiektami bazy danych - tzn. serwerami baz danych, bazami danych, alokacją, (w tym replikami i fragmentami), urządzeniami składowania danych (danymi i kopią archiwalną), grupami użytkowników i użytkownikami oraz ich prawami, tabelami z indeksami, kolumnami, regułami integralności, wyzwalaczami, widokami, procedurami- które występują (pod różnymi nazwami) we wszystkich porównanych wyżej narzędziach.

Dodano natomiast bardzo istotną funkcją, której brak w porównanych SZBD, tzn. zarządzanie fizyczną alokacją baz częściowych: fragmentów i replik. Dane potrzebne do zastosowana algorytmów zbiera się środkami SZBD. Założono, że zarządzanie powinno odbywać się na podstawie monitorowania statystyk serwerów i obiektów bazy danych.

Zaprojektowane narzędzie daje też możliwość uwzględniania wiedzy i doświadczenia administratora w celu weryfikacji i modyfikowania otrzymywanych wyników fizycznej fragmentacji bazy danych.

Literatura

[1] Ozsu M.T., ***Distributed Database Management***, Prentice Hall Inc., 1994.

[2] McCprmick W.T., Schweitzer P.J., White T.W., ***Problem Decomposition and Data Reorganization by a Clastering Technique***, 1972.

[3] Rennhackkamp M., ***Backup and Recovery***, DBMS online, March 1997.

[4] Peter Rob P., Coronel C., ***Database Systems - Design, Implementation and Management***, Boyd & Fraser Publishing Company, 1995.

[5] Rennhackkamp M., ***Server Administration***, DBMS online, January 1997.

Wojciech Czujowski
Silicon & Software Systems, Dublin, Irlandia
Jacek Gruber
Wydziałowy Zakład Informatyki, Politechnika Wrocławska

Pomiar niezawodności oprogramowania

Piotr Jędrzejowicz

Rolę produktywności w procesie tworzenia oprogramowania dostrzeżono wiele lat temu. Zainteresowanie jakością oprogramowania pojawiło się później. Jakość stała się istotnym problemem dla inżynierii oprogramowania w związku z dramatycznym wzrostem zapotrzebowania na systemy informacyjne oraz rozwojem technologii informacyjnych.

Popyt na systemy o złożonej architekturze sprzętowej i programowej rośnie szybciej niż możliwości ich projektowania, implementacji, testowania i utrzymywania. Nietrudno obronić tezę, że w przeciwieństwie do postępów w zakresie technologii sprzętu rozwój po stronie oprogramowania nie nadąża w zakresie wszystkich podstawowych aspektów, tj. jakości, produktywności, kosztu i efektywności.

Twórcy oprogramowania uznają dzisiaj, że jakość i produktywność są wzajemnie powiązane. W przeszłości wielu sądziło, że jakość i produktywność są przeciwstawne, a więc że wzrost wydajności jest możliwy tylko poprzez obniżenie jakości. Współcześni menedżerowie na własnej skórze odczuli, że powyższe założenie jest prawie zawsze błędne. Producent oprogramowania, który wprowadzi skuteczny program podnoszenia jakości może oczekiwać efektów trojakiego rodzaju. Po pierwsze produktywność jego zespołu wzrośnie, po drugie, łączne koszty produkcji oprogramowania zmniejszą się i po trzecie, konkurencyjność produktu i firmy wzrośnie.

Niezawodność jest uznawana prawdopodobnie, za najważniejszy atrybut jakości oprogramowania [1]. Oprogramowanie o wysokiej niezawodności posiada w większym lub mniejszym stopniu większość spośród własności uznawanych za główne cechy jakości. Dodatkowo, niezawodność okazała się tym szczególnym atrybutem jakości oprogramowania, który najłatwiej poddaje się próbom kwantyfikacji. W istocie norma ISO 9000-3 określa rejestrację uszkodzeń oprogramowania w trakcie jego użytkowania jako jedyną wymaganą miarę jakości [2]. Niezawodność oprogramowania rozumiana jest jako prawdopodobieństwo działania oprogramowania bez uszkodzeń przez określony okres i w określonym otoczeniu [3].

Ostatnio do analizy i oceny własności systemów zaczęto używać pojęcia wiarygodności (*dependability*). Wiarygodność obejmuje, między innymi, takie atrybuty jak niezawodność, gotowość, bezpieczeństwo czy naprawialność. Wyróżnia się rozmaite zagrożenia dla wiarygodności (np. błędy i uszkodzenia) oraz środki zapewnienia wiarygodności (tolerowanie uszkodzeń, usuwanie uszkodzeń, prewencja uszkodzeń) [4]. Niezawodność pozostaje podstawowym atrybutem wiarygodności. Może ona być użyteczna do:

- specyfikowania własności, które spełniać powinien system;
- oceny jakości systemu w związku z zagrożeniami i użytymi środkami zapewnienia wiarygodności.

Wytwórcy oprogramowania doskonale rozumieją potrzebę dostarczania swoim klientom niezawodnych produktów, zdając sobie jednocześnie sprawę, że poważna część ponoszonych kosztów związana jest z zapewnieniem, na wszystkich etapach cyklu rozwojowego, rozwiązań gwarantujących wymaganą niezawodność. Z powyższego wynika ogromne zapotrzebowanie na systematyczne i skuteczne zastosowania metod inżynierii niezawodności oprogramowania (*software reliability engineering*). Inżynieria niezawodności oprogramowania zajmuje się ilościową analizą zachowania się oprogramowania w trakcie eksploatacji w odniesieniu do wymagań użytkownika, dotyczących niezawodności [5].

Inżynieria niezawodności oprogramowania (INO) obejmuje pomiary niezawodności, analizę wpływu atrybutów i wielkości charakteryzujących proces projektowania, architekturę oprogramowania, implementację oraz profil eksploatacyjny na niezawodność, a także zastosowanie uzyskanej wiedzy do zarządzania procesem tworzenia oprogramowania.

Przedmiotem niniejszego wykładu jest próba ukazania, w jaki sposób oraz w jakim zakresie wyniki pomiarów niezawodności mogą być wykorzystane do lepszego zarządzania produkcją oprogramowania.

Cele programu pomiarów niezawodności oprogramowania

Kluczowym celem każdego programu pomiarów niezawodności oprogramowania jest dostarczenie niezbędnych informacji do dalszych czynności należących do INO, które prowadzą do lepszego zarządzania procesem tworzenia oprogramowania. Oczekiwane korzyści proponowanego podejścia obejmują pełniejszą identyfikację potrzeb klienta, skrócenie cyklu rozwoju produktu, zwiększenie produktywności oraz lepsze planowanie zużycia zasobów. Rezultaty pomiarów niezawodności są krytyczne dla:

- uzyskania pewności, że oprogramowanie osiągnęło wymagany poziom bezpieczeństwa;
- decyzji o zakończeniu testowania oprogramowania;

- analizy i zarządzania ryzykiem związanym z przekazaniem oprogramowania do użytkowania;
- zapewnieniem jakości oprogramowania.

Bezpośrednim celem każdego programu pomiarów niezawodności jest dostarczenie co najmniej jednej spośród poniższych informacji:

- estymacja aktualnej niezawodności oprogramowania;
- predykcja przyszłej niezawodności oprogramowania;
- porównanie wyników estymacji lub predykcji ze zidentyfikowanymi potrzebami klienta.

Określenie celów programu pomiarów niezawodności determinuje dalsze czynności wykonywane w ramach INO (identyfikacja profilu funkcjonalnego i operacyjnego, klasyfikacja uszkodzeń ze względu na ich znaczenie, analiza potrzeb użytkownika w zakresie niezawodności czy specyfikacja poziomu niezawodności oprogramowania). Te czynności, z kolei, wpływają na stosowane metody, modele, zasoby oraz niezbędny budżet.

Ramy czasowe oraz metodologia pomiarów niezawodności oprogramowania

Ramy czasowe pomiarów niezawodności obejmują okres od sformułowania koncepcji produktu aż do jego użytkowania przez klienta. Czynności związane z mierzeniem niezawodności powinny towarzyszyć wszystkim fazom cyklu życia oprogramowania. Donnelly [6] formułuje następującą listę działań podejmowanych w ramach INO (czynności związane z pomiarami niezawodności zostały wyróżnione pogrubioną czcionką):

Format danych o uszkodzeniach i procedury ich zbierania

Zaprojektowanie i wdrożenie procedur zbierania danych o niezawodności wymaga podjęcia następujących kroków [7]:

- opracowanie planu obejmującego cele przedsięwzięcia oraz definiującego sposoby zbierania danych. W planie tym należy określić jak często, przez kogo i w jakim zakresie będą zbierane dane, jak będą one przetwarzane i przechowywane, jakie narzędzia programowe będą używane, jak proces zbierania danych będzie monitorowany oraz jaki wysiłek i zasoby są niezbędne do realizacji całego projektu;
- ocena dostępności, dojrzałości i użyteczności proponowanych narzędzi programowych;
- szkolenie wszystkich uczestników projektu (analitycy systemu - projektanci, programiści, personel testujący, pracownicy pionu zapewnienia jakości, menedżerowie projektu oraz użytkownicy);
- implementacja planu oraz zapewnienie dostępu do niezbędnych zasobów;
- regularne monitorowanie procesu zbierania danych;
- zapewnienie informacji zwrotnych wszystkim uczestnikom projektu (włącznie z osobami bezpośrednio zaangażowanymi w zbieranie danych).

Dla potrzeb inżynierii niezawodności oprogramowania zbierane mogą być dwa rodzaje danych o uszkodzeniach:

- dane o liczbie uszkodzeń (lub dane o liczbie uszkodzeń na jednostkę czasu);
- dane o czasach między uszkodzeniami.

Wykorzystanie modeli inżynierii niezawodności oprogramowania w trakcie testowania opiera się na założeniu pomia-

Fazy cyklu życia	Działania
Analiza opłacalności oraz specyfikacja wymagań	**określenie profilu funkcjonalnego** **definiowanie i klasyfikacja uszkodzeń** **identyfikacja potrzeb klienta w zakresie niezawodności** analizy alternatywnych rozwiązań (*trade-off studies*) **określenie celów w zakresie niezawodności**
Projektowanie oraz implementacja	alokacja niezawodności do modułów poszukiwanie rozwiązania prowadzącego do osiągnięcia celów w zakresie niezawodności koncentracja zasobów w zgodzie z profilem funkcjonalnym kontrola powstawania i propagacji błędów **pomiary niezawodności oprogramowania pochodzącego ze źródeł zewnętrznych**
Testowanie i próbna eksploatacja	**określenie profilu operacyjnego** **testowanie wzrostu niezawodności** **monitorowanie postępów procesu testowania** alokacja dodatkowych zasobów na potrzeby testowania (jeśli wystąpi taka konieczność) **certyfikacja stopnia osiągnięcia celów w zakresie niezawodności**
Utrzymanie i obsługa	planowanie niezbędnych zasobów **monitorowanie niezawodności eksploatacyjnej na tle postawionych celów w zakresie niezawodności** monitorowanie satysfakcji klienta harmonogramowanie nowych rozwiązań i ulepszeń poprzez monitorowanie niezawodności wykorzystanie miar niezawodności do lepszego sterowania procesem tworzenia oprogramowania

Tab.1. Działania podejmowane w ramach inżynierii niezawodności oprogramowania

rów czasów obliczeń, w trakcie których występują uszkodzenia. Czas pracy CPU jest bezpośrednią miarą czasu obliczeń. Oprócz danych o uszkodzeniach potrzebne są dodatkowe informacje. Mogą one obejmować m.in. klasę dotkliwości danego uszkodzenia, informacje o wykorzystywanych zasobach systemu komputerowego, liczbę nowych lub zmodyfikowanych instrukcji kodu źródłowego, średni czas wykonywania takiej instrukcji, wielkość programu mierzoną liczbą instrukcji kodu wynikowego, identyfikator testowanej funkcji, czas trwania oraz datę aktualnej sesji testowania. Dalsze dane będą potrzebne do analiz dotyczących alternatywnych rozwiązań programowych.

Wykorzystanie modeli INO w trakcie użytkowania, obok rejestracji uszkodzeń oraz czasów ich napraw, wymaga dodatkowych informacji [8]. Zbierane dane mogą obejmować między innymi czas wystąpienia uszkodzeń, datę otrzymania informacji o uszkodzeniu, identyfikator klienta oraz instalacji, klasę dotkliwości uszkodzenia oraz okres degradacji, przyczyny uszkodzenia - bezpośrednie i pierwotne, poziom, na którym wystąpiło uszkodzenie (system, podsystem), aktualny status problemu oraz procedury jego rozwiązywania. Dodatkowo zbierane mogą być informacje o wersji oprogramowania, typie uszkodzenia (błąd danych lub błąd programu), opis uszkodzenia lub dane wejściowe reprodukujące uszkodzenie.

W wypadku oprogramowania użytkowanego równolegle w wielu wersjach oraz w wypadku programów „z półki" lub po prostu w wypadkach kiedy kontekst użytkowania programu przez klienta tak dyktuje, dane o uszkodzeniach rejestrowane są tylko w dziedzinie czasu kalendarzowego. Zazwyczaj bezpośrednie monitorowanie intensywności użytkowania takiego oprogramowania jest niemożliwe bądź mało praktyczne.

Wymagane zasoby oraz koszty pomiarów niezawodności

Programy pomiarów niezawodności oraz niektóre pozostałe czynności charakterystyczne dla inżynierii niezawodności oprogramowania mogą być włączane do procesów rozwoju oprogramowania stopniowo na bazie kolejnych projektów lub mogą być traktowane jako całościowe rozwiązanie strategiczne obejmujące pełne spektrum działań podejmowanych w ramach INO. Obydwa podejścia wymagają odpowiedniego zaplanowania i alokacji potrzebnych zasobów. Zasoby, o których mowa, obejmują, z zasady, wiedzę (*know-how*), pracę i czas. Obydwa podejścia do wdrażania metod inżynierii niezawodności oprogramowania wymagają także przygotowań, zaangażowania oraz motywacji na różnych szczeblach organizacyjnych. Pogląd na niezbędną wielkość angażowanych środków dają informacje przedstawione w tabeli 2.

Bezpiecznym założeniem dla większości projektów jest przyjęcie, że koszty działań związanych z inżynierią niezawodności oprogramowania nie przekroczą 1,5% - 2% wszystkich pozostałych kosztów.

Wdrażanie programu mierzenia niezawodności oprogramowania

Mierzenie niezawodności oprogramowania jest logicznym pierwszym krokiem wdrażania w organizacji działań w ramach inżynierii niezawodności oprogramowania. Sukces programu mierzenia niezawodności zależy od zaangażowania i motywacji kierownictwa najwyższego szczebla. Program początkuje zwykle przygotowanie i przedstawienie kierownictwu firmy syntetycznej informacji o celach i założeniach przedsięwzięcia. Z kolei przepro-

Projekt	Czynność	Koszt
Projekt realizowany przez 10-osobowy zespół, około 100000 wierszy kodu źródłowego, czasokres trwania projektu - 18 miesięcy [6]	• transfer wiedzy z zakresu INO • identyfikacja profilu operacyjnego • zbieranie i analiza danych testowych	• 1 osoba/miesiąc (konsultant zewnętrzny) • około 2 osobomiesięcy • około 0.3 osobomiesięcy
Bardzo duży projekt (zespół ponad 1000 osób, wiele różnych dużych profili operacyjnych) [6]	• pełne spektrum działań INO	• jedna osoba, pełen etat
Średni projekt „System informacyjny dla małej firmy", 80 000 linii kodu źródłowego, czasokres trwania projektu 12 miesięcy [9]	• szkolenie i przygotowania • identyfikacja profilu funkcjonalnego i operacyjnego • zbieranie i analiza danych o uszkodzeniach w trakcie testowania • zbieranie i analiza danych o uszkodzeniach w trakcie użytkowania przez 1 rok kalendarzowy	- 1 osoba/miesiąc • 1 osoba/miesiąc • około 2 osobomiesięcy • około 2 osobomiesięcy
Duży projekt (system operacyjny czasu rzeczywistego, użytkowany w wielu wersjach) [9]	• zbieranie danych o uszkodzeniach w trakcie użytkowania przez 3 lata kalendarzowe • analiza danych eksploatacyjnych	• około 6 osobomiesięcy • 1,5 osobomiesięcy (konsultant zewnętrzny)

Tab. 2. Koszty wybranych projektów INO

wadzana jest seria szkoleń. Początkuje ją krótki kurs o charakterze przeglądowym dla kierowników projektów softwarowych oraz pogłębiony kurs z zakresu INO dla członków zespołu odpowiedzialnego za pomiary. Równolegle lub bezpośrednio po zakończeniu tych kursów celowe jest przeprowadzenie warsztatu planistycznego, na którym formułuje się wstępny plan działań wdrożeniowych. W tej fazie implementacji organizacja może odnieść korzyści z zatrudnienia i wykorzystania wiedzy konsultanta zewnętrznego.

Dalsze przygotowania i planowanie obejmują podjęcie decyzji o sposobie zbierania danych o uszkodzeniach (automatyczne lub ręczne). Konsekwencją powyższego jest decyzja o alokacji odpowiedniego personelu i zasobów niezbędnych do wykonania postawionych zadań. Na tym etapie dokonywana jest wstępna analiza danych o uszkodzeniach, która pozwala wyeliminować informacje nadmiarowe i nieistotne. W efekcie projektowane są procedury filtracji i wstępnego przetwarzania danych o uszkodzeniach, które pozwalają na uzyskanie istotnych informacji w formacie umożliwiającym dalsze analizy.

W ramach czynności związanych z wdrażaniem programu mierzenia niezawodności podejmowane są także decyzje co do wyboru narzędzi analitycznych dla całego projektu. Decyzje w powyższym zakresie uwzględniają dostępność, koszt przygotowania, zakupu oraz instalacji i utrzymania, liczbę projektów softwarowych, dla których będą prowadzone pomiary, rodzaje tworzonych systemów, przyjazność narzędzi, harmonogram projektu oraz rodzaj zbieranych danych, jak również możliwości analityczne i inne cechy narzędzi INO.

W ramach działań wdrożeniowych podejmowana jest także decyzja o wstępnym wyborze modeli oraz sposobach ich wykorzystania. Po wyborze modelu jego jakość i przydatność mogą być oceniane metodami analitycznymi. Wstępny wybór modeli opiera się raczej na innych kryteriach (doświadczenia z podobnych projektów mogą tu być dostateczną przesłanką).

Weryfikacja i walidacja programu mierzenia niezawodności oprogramowania

Weryfikacja programu mierzenia niezawodności oprogramowania polega na zapewnieniu, że metody i techniki używane do zbierania danych o uszkodzeniach, a następnie do analizy tych danych w celu otrzymania predykcji i estymacji, zostały zastosowane w poprawny sposób. Weryfikacja dotyczy oceny procedur i działań w aspekcie zapewnienia, że otrzymane rezultaty są prawdziwe, zaś wnioski zostały wysnute w sposób właściwy. Kroki związane z weryfikacją pomiarów niezawodności powinny zostać precyzyjnie zaplanowane już na etapie przygotowania projektu. Realizacja tych kroków musi być monitorowana na poziomie kierownictwa projektu.

Walidacja, z kolei, obejmuje zapewnienie, że wybrano odpowiednią metodologię i wdrożono efektywne rozwiązanie. Jest to zadanie złożone. W wypadku niezawodności oprogramowania należy nie tylko zademonstrować empirycznie spełnienie warunku reprezentatywności [10] oraz wiarygodności otrzymanych predykcji, ale także przekonać, że proporcje pomiędzy korzyściami a kosztami projektu nie mogły być bardziej korzystne. Walidacja obejmuje m.in. ocenę dokładności stosowanych modeli, błędów predykcji oraz stacjonarności analizowanych danych o uszkodzeniach.

❖ ❖ ❖

Chociaż działania wymagane w trakcie implementacji, a potem realizacji programu mierzenia niezawodności oprogramowania, wydają się względnie proste i dobrze zdefiniowane, to jednak istnieje wiele czynników, które mogą zmniejszyć szanse powodzenia całego programu. Pośród nieprzyjaznych czynników egzogenicznych, których wystąpienie powinno być brane pod uwagę, wymienić należy:

- niestacjonarność danych dotyczących niezawodności oprogramowania;
- zmienne schematy użytkowania oprogramowania;
- zmienność profilu funkcjonalnego aplikacji;
- zmienność otoczenia programowego.

Pośród nieprzyjaznych czynników endogenicznych do istotnych należą następujące:

- brak motywacji i zaangażowania ze strony personelu odpowiedzialnego za rozwijanie oprogramowania. Przyczyną powyższego może być obawa, że estymacje niezawodności posłużą do oceny jakości pracy twórców oprogramowania. Inną przyczyną jest nieznajomość metodologii INO;
- niechęć ze strony kierownictwa firmy do rozpowszechniania informacji o niezawodności własnych produktów softwarowych, a w szczególności do udostępniania takich informacji konkurencji.

Pomimo obecności wspomnianych czynników inżynieria niezawodności oprogramowania jest obecnie dojrzałą dziedziną inżynierską, która dysponuje sprawdzonymi metodami, narzędziami oraz schematami działań.

Literatura

[1] Musa J.D., Iannino A., Okumoto K., ***Software Reliability. Measurement, Prediction, Application***, McGraw-Hill, New York 1987.

[2] ***Quality Management and Quality Assurance Standards - Part 3: Guidelines for the Application of ISO 9001 to the Development, Supply and Maintenance of Software***, ISO 9000-3, 1991.

[3] ***Standard Glossary of Software Engineering Terminology***, STD-729-1991, ANSI/IEEE, 1991.

[4] Laprie J.-C., ***Dependability - Its Attributes, Impairments and Means***, in: B.Randell, J.-C.Laprie, H.Kopetz, B.Littlewood (eds.), Predictably Dependable Computing Systems, Springer, Berlin 1995, pp. 3 - 24.

[5] ***Charter and Organization of the Software Reliability Engineering Committee***, IEEE, New York 1995.

[6] Donnelly M., B.Everett, J.Musa, G.Wilson, ***Best Current Practice of SRE***, in: M.R. Lyu (ed.), Handbook of Software Reliability Engineering, IEEE Computer Society Press, McGraw - Hill, New York 1996, pp. 219 - 254.

[7] Nikora A.P., M.R.Lyu, ***Software Reliability Measurement Experience***, in: M.R. Lyu (ed.), Handbook of Software Reliability Engineering, IEEE Computer Society Press, McGraw - Hill, New York 1996, pp. 255 - 302.

[8] ***Guide for the Use of IEEE Standard Dictionary of Measures to Produce Reliable Software***, IEEE Std-982.2-1988, IEEE, New York 1988.

[9] Jędrzejowicz P., E,Rudolph, ***Experiences in software product assessment***, Proc. 8th European Software Control and Metrics Conference, Berlin 1997, pp. 150 – 155.

[10] Fenton N.E., ***Software Measurement: A Necessary Scientific Basis***, in: B.Randell, J.-C.Laprie, H.Kopetz, B.Littlewood (eds.), Predictably Dependable Computing Systems, Springer, Berlin 1995, pp. 67 - 84.

Piotr Jędrzejowicz jest pracownikiem Katedry Informatyki Wyższej Szkoły Morskiej w Gdyni
E-mail: pj@wsm.gdynia.pl

Podstawowe elementy krytyki reengineeringu i analiza ich zasadności

Janusz Wielki

Niemal od samego początku reengineering zbierał różne opinie krytyczne pod swoim adresem. Ukazywały się one najczęściej w specjalistycznych publikacjach i dotyczyły różnych aspektów wiążących się z tą koncepcją.

W czerwcu 1997 roku bardzo spektakularny „atak" na reengineering przypuścił wpływowy i poczytny „Business Week" piórem swego analityka Johna A. Byrna. Opierając się mianowicie na wynikach badań Bain & Co. Consultants, stwierdził on wszem i wobec, iż reengineering jest koncepcją odnoszącą zdecydowanie najmniejsze sukcesy spośród wszystkich najpopularniejszych koncepcji zarządzania związanych z wprowadzaniem zmian w funkcjonowaniu organizacji. Dodatkowym „smaczkiem" całej sprawy jest fakt, iż ten sam „Business Week" parę lat wcześniej nazwał współtwórcę reengineeringu, Michaela Hammera, *jednym z czterech najwybitniejszych guru zarządzania lat dziewięćdziesiątych* ([1]). Ciekawe jest wobec tego, co takiego zdarzyło się podczas tego, bądź co bądź, krótkiego okresu, iż ten wpływowy magazyn zmienił w tak radykalny sposób opinię o reengineeringu i jego współtwórcy. Oczywiste jest również, iż jeśli opinia wyrażona przez „Business Week" byłaby rzeczywiście prawdziwa, to nie byłoby specjalnego sensu zajmować się tą koncepcją w przyszłości. Wydaje się również, iż ocena wartości reengineeringu byłaby bardzo istotna dla polskich menedżerów ze względu na fakt, iż wiele krajowych firm wymaga przeprowadzenia głębokich i radykalnych zmian, a reengineering jest wskazywany w niektórych publikacjach jako właśnie ta koncepcja, która jest najbardziej użyteczną do tego celu. Przykładem może tu być artykuł S.Trautmana, w którym autor twierdzi, iż umiejętność wykorzystania reengineeringu w praktyce jest niezbędna dla krajowych organizacji w warunkach globalnej konkurencji i że stanowi on *skok jakościowy i wyzwanie dla dyrekcji i zarządów większości polskich firm*[1] [2]. Dlatego też w dalszej części spróbujemy, opierając się na dostępnych źródłach, dokonać przeglądu podstawowych zarzutów pod adresem reengineeringu i ocenić ich zasadność.

Najczęstsze zarzuty pod adresem reengineeringu

Tak jak wspomniano wcześniej spora liczba krytycznych uwag w odniesieniu do reengineeringu pojawiła się w literaturze tematu. Poniżej przedstawionych zostało sześć najczęściej powtarzających się spośród nich i kolejno zostaną one omówione.

Bardzo duży procent projektów związanych z reengineeringiem kończy się niepowodzeniem

Jest to zdecydowanie najpowszechniejszy i chyba najbardziej znaczący zarzut w stosunku do BPR. Krytycy reengineeringu wskazują na fakt, iż około 70 % prób implementacji reengineeringu kończy się niepowodzeniem (np. [3], [4]). Tak naprawdę to zarzut ten nie jest specjalnie zaskakujący, gdyż już Hammer i Champy pisali w swej książce: *nasze amatorskie szacunki wskazują, iż około 50% do 70%, organizacji, które podjęły wysiłek związany z reengineeringiem, nie osiągnęły takich rezultatów, jakich zakładały* [1]. Wydaje się zresztą, iż wyjaśnienie tego faktu jest stosunkowo proste i oczywiste. Ryzyko wiążące się z implementacją BPR jest znaczne i na fakt ten wskazywano wiele razy ([6], s. 189).

Ryzykowność reengineeringu wynika głównie z zakresu działań, które muszą być podjęte. Obejmują one zmiany w bardzo wielu obszarach [7]:

- podsystemy zarządzania (wartości, wskaźniki , styl pracy);
- czynniki ludzkie (zawody, aspekty kulturowe, umiejętności);
- technologia informatyczna;
- struktura organizacyjna (kompresja pionowa).

Ten duży zakres koniecznych zmian, jak również ich różnorodność, powodują, iż zwiększa się prawdopodobieństwo popełnienia błędu czy też niedoceniania wpływu pewnych istotnych elementów, czego rezultatem może być niepowodzenie całego przedsięwzięcia. Oczywiste jest, iż tak wysoki poziom ryzyka kompensowany jest korzyściami, które można osiągnąć w wypadku powodzenia projektu. Potencjalne korzyści są znacznie większe niż te będące rezultatem zastosowania koncepcji takich jak *business process improvement* czy TQM. W obydwu z nich ryzyko niepowodzenia jest znacznie niższe z uwagi na fakt, iż ich celem jest jedynie doskonalenie procesów w istniejącej postaci, a nie ich dogłębne przeprojektowanie. Spróbujmy się przyjrzeć, jakiego rodzaju elemen-

[1] Pierwsze kroki zostały poczynione w Polsce dość wcześnie, gdyż w 1994 roku. Wtedy to zakończony został w Stomilu Sanok S.A. pionierski, jeżeli chodzi o kraje byłego bloku wschodniego, projekt reengineeringu [5].

ty decydują najczęściej o niepowodzeniu reengineeringu. Wyniki badań przeprowadzonych w 1995 roku przez „CFO Magazine" wspólnie z *Ernst & Young* pośród menedżerów amerykańskich wskazują, iż za elementy w największym stopniu decydujące o niepowodzeniu całego projektu uważają oni [8]:

- niewłaściwe kierowanie przebiegiem projektu 17,4%
- czynniki natury kulturowej 15,5%
- brak wsparcia dla projektu ze strony kierownictwa 4,7%

Z kolei według badań „1996 CIO Survey" przeprowadzonych rok później przez Deloitte & Touche Consulting Group najważniejsze przyczyny porażek stanowią [9]:

- opór wewnątrz organizacji 82%
- nieodpowiednie wspomaganie ze strony kierownictwa 72%
- nierealistyczne oczekiwania 66%
- nieodpowiednie kierowanie projektem 60%.

Warto do tego dodać, iż szczególnie we wczesnym okresie BPR był po prostu modny i wielu szefów firm decydowało się na przeprowadzenie reengineeringu bez dokonania oceny rzeczywistej jego potrzeby, podążając za ogólnym trendem [10]. Natknąć się również można na stwierdzenie, iż tak wysoki poziom niepowodzeń jest rezultatem włączania do szacunków projektów, które określane co prawda były mianem reengineeringu, ale tak naprawdę nic wspólnego z nim nie miały [2].

Obok powszechnie cytowanych danych o niepowodzeniach dotykających 70% projektów reengineeringu znaleźć można również dane wskazujące na całkiem znaczny poziom ogólnego zadowolenia menedżerów z rezultatów osiąganych przy wykorzystaniu BPR. Przykładem mogą być wyniki cytowanych wcześniej badań „1996 CIO Survey" (tabela 1).

Satysfakcjonujące	76 %
Trudno powiedzieć	13 %
Niezadowalające	11 %

Tabela 1. Ocena rezultatów osiągniętych dzięki wykorzystaniu reengineeringu
Źródło: [9]

Również wyniki badań przeprowadzonych pośród 30 firm meksykańskich (należących do 500 największych w tym kraju), które wprowadzały w życie program reengineeringu, wskazują na znaczny poziom zadowolenia z osiągniętych rezultatów (tabela 2).

[2] Bliżej odniesiemy się do tego zagadnienia w podsumowaniu.

Wartościowe rezultaty	69 %
Przeciętne rezultaty	21 %
Nie czas jeszcze na ocenę rezultatów	10 %

Tabela 2. Ocena rezultatów reengineeringu w 30 firmach meksykańskich
Źródło: [11]

Niedocenianie roli pracowników w transformacji procesów

Jest to poważny i generalnie rzecz biorąc słuszny zarzut. Reengineering dość powszechnie zresztą zyskał sobie opinię koncepcji, która koncentruje się głównie na technologii informatycznej, traktując pracowników jako drugorzędny element, którym można dowolnie manipulować. Bardzo dosadnie ujął tę sytuację T. Davenport, pisząc, że *BPR traktował ludzi pracujących wewnątrz firmy tak, jakby byli oni po prostu kolejnymi bitami i bajtami, wymiennymi częściami, które muszą zostać poddane reengineeringowi* [12]. Prawda jest natomiast zupełnie inna. Ludzie stanowią kluczowy element każdego programu reengineeringu, ponieważ to właśnie oni wprowadzać będą zmiany, a następnie będą musieli funkcjonować w nowych dla siebie rolach, jak również warunkach. Dlatego też docenienie znaczenia powyższego faktu przez menedżerów oraz ich pomoc wydają się być niezwykle istotnymi czynnikami.

Konsekwencją niedoceniania roli pracowników w reengineeringu stało się również niedoszacowanie poziomu ich obaw związanych z nową dla nich sytuacją. Obawy te spowodowane są wieloma problemami. Jako że projekty BPR są zwykle dosyć złożonej natury, pracownicy muszą w związku z tym stawić czoła wielu nowym problemom. Pierwszym z nich jest odmienny styl pracy. W organizacjach, w których przeprowadzono reengineering, mają oni zdecydowanie większy zakres odpowiedzialności, muszą podejmować decyzje, jak również oczekuje się od nich elastyczności w działaniu. Wydaje się, iż w początkowym okresie może to być dla niektórych z nich dosyć trudne.

Kolejnym trudnym do przezwyciężenia elementem są zmiany w odniesieniu do preferowanych w firmie wartości i poglądów. Ich tworzenie może być tak naprawdę najtrudniejszym aspektem do przezwyciężenia, ponieważ powstają one przez lata i w związku z tym trudno oczekiwać, iż można je będzie zmienić jednym posunięciem. Również zmiana nastawienia do technologii informatycznych, szczególnie w wypadku starszych pracowników, jest ważnym i trudnym elementem wprowadzanych zmian.

Kolejnym elementem, który może być poważnym źródłem obaw pracowników, jest konieczność opanowania przez nich nowych umiejętności oraz kompetencje. Będą one oczywiście różne w zależności od pozycji zajmowanej w hierarchii organizacyjnej.

Zaliczyć do nich można między innymi [13]:

- umiejętność zarządzania ryzykiem;
- umiejętność rozwiązywania problemów;
- umiejętność pracy grupowej;
- ukierunkowanie na potrzeby klientów;
- umiejętność myślenia strategicznego;
- umiejętność planowania.

I na koniec jeszcze jeden element powiększający ich obawy względem reengineeringu. Jest nim obawa przed zwolnieniami. Jej źródłem są głównie przekazywane z ust do ust opinie, z których wynika, iż typową konsekwencją projektów BPR są masowe zwolnienia. W ten sposób możemy przejść do kolejnego zarzutu.

Reengineering jest tym samym co downsizing

Kolejna grupa krytyków reengineeringu twierdzi, iż prawie każdy projekt BPR prowadzi do zwolnień na dużą skalę. Niektórzy posuwają się o krok dalej, twierdząc, że reengineering równoważny jest *downsizingowi*, czyli „odchudzaniu" organizacji. Jest to skojarzenie zdecydowanie mylne [14]. Bywa jednak tak głęboko zakorzenione w świadomości pracowników, iż np. w firmie Corning świadomie unikano używania słowa reengineering w stosunku do przeprowadzanych zmian [15].

Dostępne dane wskazują, iż typowym rezultatem implementacji reengineeringu jest utrata pracy przez około 21% pracowników ([12], [16]). Bywa jednak, iż procent ten jest znacznie większy. Na przykład w wyniku wspomnianego wcześniej reengineeringu w Stomilu Sanok S.A. pracę utraciło 2500 ludzi z 6000 załogi, czyli 41% [5].

Konieczne jest jednak podkreślenie, iż zwolnienia same w sobie nie są absolutnie celem reengineeringu, bywają one raczej jego rezultatem [17]. Kwestię tę zresztą wyjaśnił parę lat temu sam M. Hammer w artykule opublikowanym na łamach „The Economist" ([18]).

Wydaje się jednak sprawą dosyć oczywistą, że wprowadzanie w życie głębokich i radykalnych zmian przy znacznym wsparciu technologii informatycznej może prowadzić, i często prowadzi, do eliminacji części stanowisk. Również kompresja pionowa dokonująca się w organizacjach oraz rozszerzenie zakresu uprawnień i odpowiedzialności pracowników prowadzi do eliminacji głównie menedżerów średniego szczebla. Warto jednak w tym miejscu przypomnieć, iż zmniejszanie się ilości menedżerów średniego szczebla nie jest zjawiskiem wcale nowym. Trzeba bowiem przypomnieć, iż w rezultacie wprowadzanie w życie programów TQM jedynie w USA w okresie dziesięciu lat ubyło około miliona tego typu stanowisk [19].

Również pewien wpływ na opinię o częstych zwolnieniach związanych z implementacją reengineeringu miało ścisłe stosowanie w praktyce koncepcji „czystej kartki" (*clean slate*), szczególnie we wczesnym stadium rozwoju BPR, chociaż powszechność tego zjawiska nie była zbyt znaczna [7].

Reengineering nie jest niczym istotnym, a jedynie krótkotrwałą modą

Następna liczna grupa krytyków twierdzi, iż trudno jest traktować reengineering jako koncepcję mającą przed sobą przyszłość, gdyż jest to tylko przejściowa moda. Tego typu opinie pojawiały się w licznych publikacjach prawie od zarania reengineeringu i wciąż można się natknąć na nowe [20]. Przyjrzymy się, czy taką samą opinię wyrażają na temat reengineeringu menedżerowie, czyli ludzie, którzy nie piszą na jego temat, ale mają z nim do czynienia w praktyce. Przywołajmy po raz kolejny „1996 CIO Survey". Jego wyniki wskazują jasno, iż opinie menedżerów różnią się zasadniczo od tych, które pojawią się w publikacjach prasowych. Około 84% ankietowanych menedżerów bowiem stwierdziło, iż reengineering jest *ważnym narzędziem zarządzania* (tabela 3).

Procent respondentów wyrażających określoną opinię	
Ważne narzędzie zarządzania, jeżeli jest właściwie użyte	84 %
Jeszcze jedna przejściowa moda, która prawdopodobnie zniknie za kilka lat	13 %
Głównie powód dla ludzi ze środowisk akademickich oraz konsultantów do powiększania swych dochodów	3 %

Tabela 3. Co menedżerowie sądzą na temat reengineeringu?
Źródło: [9]

Trzeba wyraźnie podkreślić, iż żaden z podstawowych komponentów reengineeringu, tj.

- technologia informatyczna;
- orientacja procesowa;
- koncepcja „czystej kartki"

nie ma cech krótkotrwałej mody, pozostają wciąż aktualne i ważne.

Znaczenie technologii informatycznych we wprowadzaniu zmian zwiększa się wciąż w kontekście gwałtownego rozwoju Internetu.

Jeszcze jedna uwaga na koniec. Do grona ludzi twierdzących, iż reengineering nabrał cech przejściowej mody, należy obecnie również jeden ze współtwórców tej koncepcji - Thomas Davenport. Chociaż docenia on znaczenie wymienionych wcześniej składników reengineringu, to jednak twierdzi, iż szczególnie w USA koncepcja ta straciła na znaczeniu. Dodaje on tego, iż BPR w zasadzie od swego początku posiadało atrybuty teorii będącej tylko przejściową moda. Według niego cechy te obejmują [21]:

- przesadne oczekiwania co do rezultatów;
- powierzchowne zrozumienie przez rzesze entuzjastów;
- posiadanie i czczenie guru;
- implementacja w firmach w uproszczony sposób.

Reengineering nie jest niczym nowym

Zarzut ten jest również dosyć często powtarzany. P. Strassmann twierdzi nawet, iż reengineering jest niczym więcej niż po prostu chwytliwą nazwą (*clever buzz word*) [22]. Tak naprawdę to rzeczywiście składniki reengineeringu, które uprzednio wymieniliśmy, były wcześniej wykorzystywane jednak zawsze osobno. Pomysł w koncepcji BPR jest odmienny. Łączy ona wszystkie te trzy komponenty razem po to, aby w ten sposób osiągnąć znaczące rezultaty [12]. Prawdą jest również, iż reengineering posługuje się głównie technikami oraz narzędziami wykorzystywanymi już wcześniej (na przykład w TQM), jednak z drugiej strony coraz więcej pojawia się aplikacji przeznaczonych specjalnie dla BPR.

Jeżeli chodzi o techniki wykorzystywane w reengineeringu, to zaliczyć do nich można na przykład *benchmarking*, analizę „ości ryby", diagram Pareto, budżetowanie, diagramy przepływów, analizę łańcucha wartości czy burzę mózgów. Z kolei przykładami wykorzystywanych narzędzi mogą być następujące aplikacje: ABC Flow Charter, Microsoft Project, PACE, Oracle Process Manager. W [7] zostały zidentyfikowane 72 techniki i 68 narzędzi wykorzystywanych podczas reengineeringu.

Teoria i praktyka reengineeringu pełna jest rewolucji i przemocy

Zarzut ten został sformułowany przez wspomnianego wcześniej P. Strassmana ([23]) i jest on w pewien sposób zasadny, jednak tylko w odniesieniu do reengineeringu w ujęciu Michaela Hammera, przedstawionym w artykule *Don't automate, obliterate* i biblii reengineeringu, napisanej wspólnie z Champym książce *Reengineering the Corporation*. Akcentował on w nich „radykalność" reengineeringu jako jego najważniejszą cechę. Jednak dosyć szybko i Hammer zaczął przesuwać akcenty w swym spojrzeniu na BPR. Już w drugim wydaniu *Reengineering the Corporation* z 1995 roku oraz w wydanej rok później *Beyond Reengineering* Hammer wskazuje, iż orientacja procesowa, a nie radykalność, powinna być widziana jako najważniejsza cecha reengineeringu [24].

Zupełnie odmienną wizję BPR prezentował w swych artykułach, a przede wszystkim w książce *Process Innovation* Thomas Davenport. Sposób przedstawienia koncepcji reengineeringu jest w jego wydaniu pozbawiony jakichkolwiek elementów agresji i jest bardzo rzeczowy. Kładzie on nacisk również na inne elementy stymulujące reengineering poza technologią informatyczną, takie jak czynnik ludzki. Niestety, spojrzenie Hammera stało się dominujące, tak że większość menedżerów buduje na jego podstawie swe wyobrażenie o reengieeringu.

❖ ❖ ❖

Wydaje się, iż powyższy przegląd najważniejszych obszarów krytyki reengineeringu prowadzi do wielu istotnych wniosków płynących na przyszłość, jednak tak naprawdę żaden z nich nie podważa znaczenia koncepcji BPR. Reengineering jest ciągle młodą, rozwijającą się i ewoluującą metodą wprowadzania radykalnych zmian w funkcjonowaniu organizacji. Pomimo iż niektórzy z jego współtwórców odsunęli się od niego (Davenport), z pewnością nie jest on koncepcją martwą, jak chcieliby niektórzy z krytyków, a raczej jak pisze D.Coleman *jesteśmy w środku rewolucji związanej z reengineeringiem* (...) [25]. Najlepszym dowodem jest wzrastająca liczba nowych metodologii, jak również aplikacji opracowywanych specjalnie na potrzeby reen-gineeringu [7]. Liczba państw, w których BPR jest wykorzystywany, dowodzi rosnącej jego popularności na świecie. W przeprowadzonych na początku 1997 przez ProSci badaniach, dotyczących BPR, uczestniczyły organizacje z 26 krajów znajdujących się na sześciu kontynentach [26]. Z pewnością mniej jest obecnie wokół reengineeringu wrzawy i emocji, których było sporo w początkowym okresie, za to więcej rzeczowego czy wręcz naukowego podejścia. Mówi się też o wejściu BPR w drugą fazę rozwoju, oczekując, iż przyniesie ona wiele zmian na lepsze, wynikających z analizy błędów pierwszego etapu, w tym również dużej ilości zaistniałych niepowodzeń [27]. Niewątpliwie nasuwa się wiele istotnych wniosków. Wydaje się, iż dla zmniejszenia prawdopodobieństwa niepowodzenia projektu należy zwrócić uwagę na następujące elementy:

- właściwy wybór procesu (procesów), w których wprowadzane mają być zmiany;
- szczegółowe przygotowanie całego projektu;
- wybór sprawdzonej metodologii.

Również szczególnie istotnym elementem każdego projektu reengineeringu powinno być właściwe docenienie roli uczestniczących w nim ludzi i przygotowanie całego systemu motywacyjnego zachęcającego do aktywnego uczestnictwa we wprowadzanych zmianach oraz minimalizującego obawy związane z możliwą utratą pracy.

Następnym istotnym nasuwającym się wnioskiem jest konieczność traktowania reengineeringu jako formy głębokich i raczej długotrwałych zmian organizacyjnych, a nie jako koncepcji mającej służyć uzyskiwaniu znaczących rezultatów w krótkim czasie (*quick fix*) [7]. Potwierdzają to wspomniane wcześniej badania przeprowadzone przez ProSci wskazujące, iż w chwili obecnej średni czas projektu reengineeringu wynosi 19,7 miesięcy, a może on być znacznie dłuższy [26]. Na przykład w GTE Telephone Operations implementacja zmian trwała 5 lat, a ocenia się, iż w niektórych wypadkach może to być nawet 8 do 10 lat [17], [28].

Jako że reengineering bywa mylony z innymi koncepcjami, spróbujmy na koniec sprecyzować, kiedy właściwie mamy z nim do czynienia. Tak jak wspomniano wcześniej, bardzo często jest on błędnie utożsamiany z *downsizingiem*, z którym tak naprawdę BPR nie ma nic wspólnego. Generalnie rzecz biorąc, działania podejmowane w ramach *downsizingu* sprowadzają się jedynie do zmniejszenia kosztów funkcjonowania organizacji poprzez zwolnienie części pracowników, natomiast w reengineeringu chodzi jednak o coś więcej.

Jest on też często mylony z *business process improvement* ([28]), pomimo iż różnica jest między nimi wyraźna. Obie koncepcje oparte są co prawda na orientacji procesowej, jednak reprezentują one zupełnie inną filozofię. Podczas gdy BPI

koncentruje się na doskonaleniu istniejących procesów, to reengineering stara się dokonać głębokiej ich transformacji, szeroko wykorzystując do tego celu technologię informatyczną. Jednym z istotnych elementów rozróżniającym dotychczas obie koncepcje był osiągany stopień poprawy podstawowych wskaźników funkcjonowania organizacji. W wypadku *business process improvement* oczekiwano ich polepszenia o około 10%, podczas gdy rezultatem reengineeringu miał być ich wzrost o *„50%, 100% lub nawet więcej"* [6]. Natknąć się można nawet na stwierdzenie, iż z prawdziwym BPR mamy do czynienie wtedy, gdy w wyniku przeprowadzonych zmian efektywność działania organizacji zostaje podwojona [28]. Jednak przeprowadzone w 1997 roku badania wskazują, iż obecne oczekiwania menedżerów w stosunku do BPR odnośnie do poprawy podstawowych wskaźników (koszty, czas trwania procesu, wydajność) znacznie się obniżyły i kształtują się na poziomie 10% do 30% [26]. Dlatego też identyfikacja reengineeringu jedynie na podstawie osiąganych wyników może być myląca. Podstawowe znaczenie ma stosowana filozofia działania.

Na koniec dwie jeszcze uwagi. Po pierwsze wydaje się, iż jeszcze dużo pracy zostało przed reengineeringiem. Biorąc pod uwagę jedynie proces marketingu, szacuje się, iż wkrótce około 70% organizacji będzie musiało poddać go reengineeringowi, przede wszystkim ze względu na rozwój Internetu i wynikające z tego konsekwencje [29]. A co do cytowanego na początku artykułu z „Business Week" kwestionującego użyteczność tej koncepcji, warto dodać, iż wyrażona tam opinia oparta została na wypowiedzi 17,3% ankietowanych twierdzących, iż są niezadowoleni z rezultatów osiągniętych przy zastosowaniu BPR.

Literatura

[1] CHAMPY J., HAMMER M., ***Reengineering the corporation***, Nicholas Brealey Publishing; 1995.

[2] TRAUTMAN S., ***Reengineering wyzwanie dla menedżerów***, „Computerworld" 1996, październik.

[3] COVERT M., ***Successfully performing business process reengineering***, [http://www.ozemail.com.au/~ieinfo/BPR.htm], 1996.

[4] MALHORTA Y., ***Business Process Redesign: An Overview***; [http://www.brint.com/papers/bpr.htm]; 1996.

[5] Wizdom Systems, Inc; BPR in Poland: How Wizdom Lead a Polish Company on its First BPR Project; [http://wizdom-systems.com/successpoland.htm] 1997.

[6] DAVENPORT T. H., ***Process innovation - reengineering work through information technology***, Harvard Business School Press, 1993.

[7] GUHA S., KETTINGER W., TENG J., ***Business Process Change - A study of methodologies, techniques and tools***, [http://theweb.badm.sc.edu/bpr/], 1997.

[8] CFO Magazine and The Ernst & Young, ***What you really think about Reengineering***, [http://www.mediapool.com/offtherecord/cfo-bpr.html], 1995.

[9] Deloitte & Touche Consulting Group; 1996 CIO Survey - Business Process Reengineering; [http://www.dttus.com/dttus/publish/cio/bpr.htm], 1996.

[10] ***Business Process Re-engineering Mailing List BPR-L***, BPR-L Digest # 1162; [BPR-@TWI.TUDELFT.NL],1997.

[11] ***Business Process Re-engineering Mailing List BPR-L***, BPR-L Digest # 883; [BPR-L@TWI.TUDELFT.NL],1996.

[12] DAVENPORT T.H., ***The fad that forgot people***, [http://www.fastcompany.com/01/reengin.html], 1996.

[13] CORRIGAN S., ***Human and organisational aspects of business process reengineering***,[http://bprc.warwick.ac.uk./shef-summ.html],1996.

[14] Business Architects, ***What Is Reengineering?***, [http://busarch.com/papers/], 1998.

[15] ACKERMAN R., NEILSON G., ***Partnering for Results: A Case Study of Re-engineering***,the Corning Way; [http://www.strategy-business.com/casestudy/96205/page1.html], 1996.

[16] FRANCIS A., MACINTOSH R., ***The market, technological and industry contexts of business process re-engineering in UK business***, [http://bpcr.warwick.ac.uk/glasgow1.html], 1996.

[17] LEE C., ***Process Re-engineering at GTE: Milestones on a Journey Not Yet Completed***; [http://www.strategy-business.com/casestudy/96406/page1.html]; 1996.

[18] HAMMER M., ***Hammer defends re-engineering***; The Economist; November 5th 1994.

[19] LUCEY T., ***Management Information Systems***, DP Publications; 1995.

[20] BYRNE J., ***Management theory - or fad of the month?***, „Business Week" 1997, June.

[21] DAVENPORT T.H., ***Prywatna korespondencja z prof. Davenportem pocztą elektroniczną***, 29 września 1997.

[22] STRASSMANN P.; The Hocus-Pocus of Reengineering; [http://www.strassmann.com/pubs/hocus-pocus.html]; 1994.

[23] STRASSMANN P.: ***Reengineering***; [http://www.strassmann.com/pubs/reengineering.html]; 1995.

[24] HAMMER M., ***Beyond Reengineering: How the Process-Centered Organization is Changing Our Work and Our Lives***, HarperCollins, 1997.

[25] COLEMAN D., ***Is re-engineering still relevant ?***, „Computerworld" 1997, 21 kwietnia, [http://cwlive.cw.com:8080/home/print9497.nsf/All/SL0421dc].

[26] ProSci; Best Practices Report for Reengineering and Business Process Design Teams - 1997 Benchmarking Study; [http://www.prosci.com]; 1997.

[27] NISSEN M., ***A Focused Review of the Reengineering Literature: Expert Frequently Asked Questions (FAQs)***; „Quality Management Journal" 3:3, 1996.

[28] BERZTISS A., ***Software Methods for Business Reengineering***, New York, Springer-Verlag; 1996.

[29] KOMENAR M., ***Electronic Marketing***; John Wiley & Sons, Inc., 1997.

Janusz Wielki jest pracownikiem Akademii Ekonomicznej we Wrocławiu
jwielki@polo.po.opole.pl

Miary oprogramowania

Andrzej Kobyliński

Pomiar oprogramowania

W naukach przyrodniczych pomiary mają tradycje sięgające tysiącleci. W czasach nowożytnych standardowa miara długości - metr - została ustanowiona w 1799 roku, chociaż nadal w wielu krajach powszechnie są wykorzystywane inne jednostki. Podobnie, wiele problemów było związanych z ujednoliceniem pomiaru temperatury. G. D. Fahrenheit wprowadził swoją skalę w 1725 roku, A. Celsius w 1742 roku, pamiętać również trzeba o skali Kelvina (1848). Z wieku XIX, okresu bujnego rozwoju nauk przyrodniczych, pochodzą często cytowane wypowiedzi koryfeuszy ówczesnej nauki:

Nauka zaczyna się wtedy, kiedy zaczyna się mierzenie (D. I. Mendelejew)

czy też:

... kiedy można zmierzyć to, o czym się mówi i wyrazić to przy pomocy liczb, to wie się coś o tym; lecz jeśli nie można tego zmierzyć, kiedy nie można tego wyrazić przy pomocy liczb, wiedza o tym jest uboga i niewystarczająca... (W. Thomson (Lord Kelvin)).

Efektowne, chociaż złośliwe jest stwierdzenie Robertsa [8], który w swej książce o teorii pomiarów napisał:

Podstawową różnicę między „dobrze rozwiniętymi" naukami, takimi jak fizyka a nieco słabiej „dobrze rozwiniętymi" naukami, takimi jak psychologia i socjologia, stanowi stopień, w jakim wykorzystywane są pomiary".

Pomiar oprogramowania jest ważnym tematem, zaliczanym do inżynierii oprogramowania. Inżynieria oprogramowania, z założenia, łączy w sobie dyscypliny informatyczne z naukami ekonomicznymi, społecznymi, o zarządzaniu. Mimo że łączy w sobie tak odległe dziedziny, chcielibyśmy, aby (zgodnie z przyjętą dla tej dyscypliny nazwą) stosować w niej metody inżynierskie, a nie czystą intuicję. Dlatego nie ulega wątpliwości, że stosując pomiary, zarówno procesu tworzenia programowania, jak i produktów programowych, można przybliżyć tę dyscyplinę do grupy nauk ścisłych.

Tak więc pomiar oprogramowania może dostarczyć nam danych do ilościowych porównań i ocen zarówno samego oprogramowania, jak i procesu jego wytwarzania.

Miary (metryki) a modele oprogramowania

Miary (metryki) oprogramowania są stosowane już od 30 lat. Prawdopodobnie pierwsza praca dotycząca miar (brak w niej odwołań do wcześniejszych prac) ukazała się w 1968 roku i dotyczyła próby pomiaru jakości [9]. W rzeczywistości tradycje stosowania miar są jeszcze dłuższe. W omawianym okresie powstało wiele miar, odnoszących się do różnych faz życia produktu.

Niektóre z proponowanych w literaturze przedmiotu miar są obiektywne (algorytmiczne), tzn. wartości tych miar dadzą się obliczyć dokładnie, zgodnie z przyjętym algorytmem, a uzyskiwane wartości są takie same, niezależnie od czasu, miejsca i osoby dokonującej pomiaru.

Nie wszystkie opisywane miary dadzą się obliczyć w tak precyzyjny sposób. Niektóre miary mają charakter subiektywny. Przede wszystkim są to te miary, które otrzymuje się w wyniku osobistej oceny pewnej cechy produktu, dokonywanej przez autora, jego zwierzchnika, eksperta, klienta czy użytkownika. Miary tego typu obarczone są wieloma wadami, ale jak inaczej wyrazić choćby „przyjazność oprogramowania", jeśli nie w sposób subiektywny?

Bardzo często miary oprogramowania nie są używane w czystej postaci, lecz wiążąc ze sobą kilka miar, tworzy się pewien model interesującej nas charakterystyki programu lub aspektu procesu programowania. W podobny sposób robi, przykładowo, importer gazu płynnego, który konstruuje sobie matematyczny model wyjaśniający (lub prognozujący) zużycie gazu, w którym to modelu zmiennymi objaśniającymi mogą być takie dane, jak stan i tempo przyrostu samochodów napędzanych gazem płynnym, pora roku (wahania sezonowa związane ze zmianami zapotrzebowania na gaz do butli turystycznych), akcje promocyjne zachęcające do ogrzewania gazowego itp. W wypadku tworzenia modeli związanych z rozwojem oprogramowania parametrami, używanymi jako zmienne objaśniające, są miary (metryki) oprogramowania.

W literaturze dotyczącej pomiarów oprogramowania pojęcia „miary" i „metryki" są traktowane jako synonimy i używa się ich wymiennie. Nie jest to całkiem ścisłe podejście. Z teorii pomiarów wynika, że aby używać terminu „metryka", jest wymagane zaistnienie określonych warunków, które nie zawsze są spełnione [11]. Stosowanie terminu „metryka" w dziedzinie pomiarów oprogramowania stanowi zatem nadużycie. Dlatego też w dalszej części systematycznie stosowany będzie termin „miara".

Miary produktu i miary procesu

Dokonując ogólnej klasyfikacji miar, zwykle dzieli się miary oprogramowania na dwie podstawowe grupy. Są to miary produktu i miary procesu programowania.

Do pierwszej grupy zaliczane są te wszystkie miary, które odnoszą się do dzieła, będącego efektem pracy analityków, projektantów i programistów.

Miary procesu próbują opisać sam proces powstawania produktu, a także otoczenie, w którym produkt powstaje. Do miar tego typu można zaliczyć, przykładowo, doświadczenie zespołu w posługiwaniu się pewnym narzędziem (mierzone latami lub miesiącami), długość stażu pracy programisty w danej firmie itp.

Przykłady kilku innych miar produktu i procesu zostaną krótko przedstawione w dalszej części.

Główne obszary zastosowań pomiarów oprogramowania

Zazwyczaj wyróżnia się sześć podstawowych obszarów zastosowań pomiarów oprogramowania. Zanim zostaną one wymienione, trzeba podkreślić, że ich zakresy niekiedy pokrywają się. Miary produktywności, jakości, a szczególnie miary złożoności, są do siebie bardzo zbliżone.

1. Modele estymacji kosztu
2. Miary produktywności
3. Modele jakości
4. Modele niezawodności
5. Miary złożoności
6. Złożoność obliczeniowa (algorytmiczna).

Przedstawmy krótkie objaśnienie wymienionych obszarów zastosowań.

ad. 1. Modele te stosuje się celem predykcji kosztu produkcji oprogramowania już na wstępnych etapach cyklu życia systemu. Najlepiej znanym modelem estymacji kosztu jest COCOMO Boehma [3].

ad. 2. O ile koszt jest kategorią ekonomiczną, której jednostka miary jest dobrze zdefiniowana, to próba pomiaru produktywności jest trudniejsza. Produktywność oblicza się zgodnie z ogólnym wzorem

$$\text{produktywność} = \frac{\text{wielkość oprogramowania}}{\text{nakład pracy [osobomiesiące]}}$$

przy czym wielkość oprogramowania może być mierzona na wiele różnych sposobów, na przykład jako liczba linii kodu źródłowego, liczba punktów funkcyjnych [1] lub, jak zaproponował Halstead [4], jako pewna kombinacja operatorów i operandów w programie.

ad. 3. Osiąganie produktów wysokiej jakości jest głównym celem inżynierii oprogramowania. Nie istnieje i nie wydaje się, by była kiedykolwiek możliwa do wprowadzenia jednostka miary jakości. Na jakość składa się, wg Boehma [2], nieco zmodyfikowana w normie ISO/IEC 9126 [6]:

- niezawodność;
- przenośność;
- efektywność;
- przyjazność;
- łatwość testowania;
- zrozumiałość;
- modyfikowalność.

Idealnie byłoby oczywiście, gdyby udało się osiągnąć maksymalne wartości w każdej z wymienionych kategorii, ale jest to niemożliwe do uzyskania ze względu na ich wzajemne uzależnienia. Dość oczywiste jest, że na przykład zwiększenie przenośności i przyjazności spowoduje zmniejszenie efektywności. Ale trzeba sobie zdawać sprawę z tego, że zwiększenie każdego z parametrów jakości powoduje zwiększenie kosztu produkcji.

ad. 4. Modele niezawodności oprogramowania są statycznymi modelami, służącymi do prognozowania średniego czasu międzyawaryjnego lub oczekiwanego czasu do kolejnego uszkodzenia.

ad. 5. Złożoność programu definiuje się jako stopień trudności zrozumienia działania programu, określany wyłącznie na podstawie jego kodu źródłowego. Miary złożoności mają wiele wspólnego ze złożonością psychologiczną. Złożoność psychologiczna uwzględnia, obok złożoności kodu programu źródłowego, również czynnik ludzki oraz dostępny materiał informacyjny, inny niż sam program.

ad. 6. Złożoność obliczeniowa (algorytmiczna) odnoszona jest nie tyle do złożoności programu, co do efektywności algorytmu. Oceniany jest sam algorytm w konfrontacji z innymi algorytmami stanowiącymi rozwiązanie tego samego problemu.

Cel dokonywania pomiarów oprogramowania

Rozważając problem pomiarów oprogramowania, a dotyczy to szczególnie środowiska praktyków, odpowiedzieć trzeba na podstawowe pytanie:

Jakie mogą być korzyści ze stosowania pomiarów oprogramowania?

Miary oprogramowania mogą znaleźć zastosowanie:

1. Jako podstawa do dokonywania różnego rodzaju estymacji;
2. Do określania złożoności;
3. Do śledzenia postępu w pracach nad systemem;
4. Do analizy błędów;
5. Przy podejmowaniu decyzji o tym, czy udało się nam już osiągnąć pożądany stopień jakości systemu;
6. W eksperymentalnym uwiarygodnianiu najlepszych praktyk programowania.

Przykłady miar i modeli, mogących mieć zastosowanie w różnych fazach cyklu życia oprogramowania

W czasie produkcji oprogramowania przez cały czas trzeba mieć na uwadze i kontrolować (oczywiście w miarę możliwości) pięć podstawowych parametrów:

- wielkość (mierzoną liczbą linii kodu, punktami funkcyjnymi lub innymi metodami);
- koszt (wyrażony w jednostkach pieniężnych);
- czas trwania (w miesiącach);
- pracochłonność (w osobomiesiącach);
- jakość (najtrudniejszą do zmierzenia i nie dającą wyrazić się pojedynczą wartością liczbową).

Ale oprócz tego w każdej z faz cyklu życia systemu można stosować inne, specyficzne dla danego etapu, miary. Z upływem czasu wykorzystywane miary mogą być coraz bardziej szczegółowe, gdyż znacznie bogatszy jest materiał, którym dysponujemy. Pokrótce przedstawimy niektóre z nich.

Faza określania wymagań

- miarą tego, w jaki sposób pracuje zespół określający wymagania, jest liczba wymagań, które zmieniły się w dalszych etapach rozwoju systemu. Jeśli jest duża liczba wymagań, które musiały zmianić się przy określaniu specyfikacji, projektowaniu i później, to nie ulega wątpliwości, że trzeba przyjrzeć się pracy tego zespołu;
- jeśli w fazie określania wymagań powstał prototyp systemu, dobrze byłoby zmierzyć, które części prototypu są szczególnie często próbowane, a które z kolei są ignorowane przez wszystkich użytkowników. Jeśli na przykład okaże się, że wszyscy eksperymentatorzy co najmniej raz wypróbowali opcję D z menu głównego, a żaden z nich nie wszedł do opcji A, to oczywiste jest, że opcja D ma podstawowe znaczenie, należy poświęcić jej wiele uwagi i starać się ją dopracować i zoptymalizować; należy również wykonać badanie, czy opcja A jest naprawdę potrzebna, czy też można usunąć ją z ostatecznej wersji produktu;
- miarą wielkości, możliwą do zastosowania w fazie określania specyfikacji, jest liczba stron dokumentu specyfikacji. Jeśli tę samą metodę pomiaru organizacja zastosuje do więcej niż jednego produktu, różnice w wielkości specyfikacji mogą stanowić ważny wskaźnik w kolejnych fazach rozwoju produktu.
- miarą, która może być przydatna w prognozowaniu wielkości przyszłego produktu, jest liczba pozycji w słowniku danych. Można tu zastosować kilka wskaźników, takich jak liczba plików, procesów itp.

Faza planowania

Mając ustaloną specyfikację systemu, trzeba zaplanować koszt i czas jego produkcji. Klient chce i musi wiedzieć, ile produkt będzie go kosztować i ile czasu będzie trwało jego opracowywanie. Niestety, nie jest to prosta czynność. Zbyt wiele zmiennych należy uwzględnić, przy czym podstawową trudność stanowi konieczność uwzględnienia czynnika ludzkiego. Z badań wynika, że różnice w produktywności programistów są ogromne, a w ekstremalnych stwierdzonych przypadkach stosunek ten może wynosić 1:28 [10]. Trzeba też mieć na uwadze, że kluczowe osoby zaangażowane w projekcie mogą chcieć zmienić pracę lub mogą się im przytrafić różne wypadki losowe.

U postaw zagadnienia estymacji kosztu i czasu trwania projektu leży problem: Jak mierzyć wielkość produktu? Ponieważ jest to obszerne odrębne zagadnienie, wspomnimy tylko w tym miejscu, że do miar wielkości zaliczamy:

- liczbę linii kodu;
- rodzinę miar *software science* Halsteada [4];
- punkty funkcyjne (oryginalne [1] oraz w wielu różnych mutacjach, np. MARK II, CHECKPOINT, ASSERT-R, SPQR/20, punkty charakterystyczne).

Dwie pierwsze metody są stosunkowo mało użyteczne, gdyż ich realne wartości są możliwe do uzyskania dopiero na stosunkowo późnych etapach w cyklu życia, stąd ogromna popularność trzeciej z wymienionych tu metod.

Próbując prognozować koszt rzeczywisty produktu, zwykle stosuje się jedną z metod:

- opierając się na ocenie ekspertów, którzy na podstawie swoich doświadczeń próbują, często bardzo trafnie, dokonać oszacowania;
- oceniając przez analogię, na podstawie informacji o uprzednio zrealizowanych przedsięwzięciach i to zarówno w firmie podejmującej zadanie, jak i w innych firmach o podobnym charakterze;
- na podstawie algorytmicznych modeli estymacji kosztu - przewidywana wielkość produktu (mierzona liczbą linii lub punktów funkcyjnych) stanowi podstawę do budowy modelu prognozującego koszt produkcji systemu. Modeli tego typu powstało wiele, ale najbardziej znany jest wspominany już model COCOMO.
- *bottom-up* (wstępującą), gdy cały projekt dzieli się na mniejsze zadania, których koszt jest oszacować łatwiej.

Faza projektowania

Nie zostały do tej pory opracowane i powszechnie zaakceptowane miary, standardowo wykorzystywane w tej fazie. Należy oczywiście kontrolować pięć miar podstawowych, można też posłużyć się następującymi wskaźnikami:

- bardzo ogólną miarą wielkości produktu docelowego jest liczba modułów w systemie;
- siła więzi modułowych i mocy modułów jest miarą jakości projektu;
- w projekcie szczegółowym jako miara złożoności modułu może być użyta tzw. miara złożoności cyklomatycznej McCabe'a, wyrażana przez liczbę dróg w module i łatwa do obliczenia jako liczba podejmowanych w module decyzji binarnych + 1;
- jeżeli cały projektowany system przedstawić w formie grafu, w którym wierzchołki odpowiadają modułom systemu, a krawędzie symbolizują przepływ informacji pomiędzy modułami, to Henry i Kafura [5] proponują, by złożoność modułu obliczać zgodnie z wzorem:

$$długość \times (fan_in \times fan_out)^2,$$

gdzie: *długość* - dowolna miara wielkości modułu;
fan_in - liczba danych przekazywanych do modułu;
fan_out - liczba danych wysyłanych z modułu.

Fazy implementacji i integracji

Począwszy od fazy implementacji, dostępny staje się kod źródłowy programów. W decydujący sposób ułatwia to stosowanie miar, gdyż od tej chwili jesteśmy w stanie dokonywać pomiarów obiektywnych. Miarami, które mogą być stosowane, począwszy od tego etapu, są wszystkie miary złożoności. Są to miary obiektywne, obliczane w sposób algorytmiczny na podstawie kodu źródłowego oprogramowania. Im będzie poświęcony cały kolejny rozdział.

Miary złożoności

Na złożoność programów komputerowych mają wpływ cztery podstawowe czynniki:

- wielkość programu;
- struktura i przepływ sterowania w programie;
- sposób użycia danych;
- styl kodowania.

Zaprojektowane w minionych latach miary w większości uwzględniają wyłącznie jeden aspekt całego problemu złożoności. Oto lista najpopularniejszych miar złożoności.

➲ **Miary wielkości**
- liczba
  - linii kodu źródłowego;
  - linii kodu źródłowego bez pustych linii i komentarzy;
  - instrukcji programu;
- miary *software science* Halsteada (rodzina kilkunastu miar)[4].

➲ **Miary sterowania**
- miara złożoności cyklomatycznej McCabe'a [7];
- absolutna i względna złożoność logiczna Gilba;
- miara złożoności MIN Chena;
- miara złożoności zakresu SCOPE Harrisona i Magela;
- miara złożoności poziomu zagnieżdżenia א Piwowarskiego;
- miara liczby przecięć w tekście programu KNOTS Woodwarda, Hennella, Hedleya.

➲ **Miary danych**
- rozpiętość między odwołaniami do danych - SPAN Elshoffa;
- średnia liczba odwołań do danych w instrukcji;
- para: segment - zmienna globalna;
- miara złożoności Q Chapina.

Literatura

[1] ALBRECHT A. J., ***Measuring Application Development Productivity, Proceedings of the IBM SHARE/GUIDE Applications Development Symposium,*** Monterrey (CA), October 1979, s. 83-92.

[2] BOEHM B. W., ***Software Engineering, IEEE Transactions on Computers*** C-25 (December 1976), s.1226-1241.

[3] BOEHM B. W., ***Software Engineering Economics***, Prentice-Hall, Englewoods Cliffs, NJ, 1981.

[4] HALSTEAD M. H., ***Elements of Software Science***, Elsevier North-Holland, New York 1977.

[5] HENRY S. M., KAFURA D., ***Software Structure Metrics Based on Information Flow, IEEE Transactions no Software Engineering*** SE-7 (September 1981), s. 510-518.

[6] ***International Standard ISO/IEC 9126: Information Technology - Software Product Evaluation - Quality Characteristics and Guidelines for Their Use***, Geneva, 1991.

[7] MCCABE T. J., ***A Complexity Measure, IEEE Transactions no Software Engineering*** SE-2 (December 1976), s. 308-320.

[8] ROBERTS F. S., ***Measurement Theory with Applications do Decisionmaking, Utility and the Social Sciences***, Addison Wesley, 1979.

[9] RUBEY R. J., HARTWICK R. D., ***Quantitative Measurement Program Quality, Proceeding of the National Computer Conference***, ACM, 1968, s. 671-677.

[10] SCHACH S. R., ***Software Engineering (Second Edition)***, Irwin and Aksen, Homewood (IL) and Boston (MA) 1993.

[11] ZUSE H., ***Software Complexity. Measures and Methods***, Walter de Gruyter, Berlin, New York 1991.

Andrzej Kobyliński, Katedra Informatyki Gospodarczej
Szkoła Główna Handlowa, Warszawa
e-mail: kobyl@sgh.waw.pl

Nagrody Softargu'98

Jury konkursu Międzynarodowych Targów Katowickich Softarg'98 w składzie: dr inż. Adam Bukowy, mgr inż. Tadeusz Bojaryn, dr inż. Juliusz Czarnowski, mgr inż. Wojciech Głazek, dr Jerzy Gołuchowski, mgr Andrzej Horodeński, mgr inż. Piotr Kowalski, dr inż. Krzysztof Nałęcki, dr inż. Zdzisław Pogoda, dr Lesław Wawrzonek (przewodniczący), przyznało nagrody, których listę wraz z krótkimi uzasadnieniami podajemy poniżej:

Nagrody:

I stopnia

Wydawnictwa Prawnicze PWN Lex Media - Publikacje Elektroniczne z Krakowa za produkt: LEX POLONICA - ON LINE. Nowoczesna aplikacja internetowa o dużej użyteczności publicznej, która może mieć pozytywny wpływ na stan świadomości prawnej.

II stopnia

➲ NET Sp.z o.o. z Katowic za produkt INTERMARKET 2.9. Nowatorska (w skali kraju) aplikacja do obsługi i organizacji handlu elektronicznego. Duża i szybko rosnąca liczba wdrożeń;

➲ Magellan Software Polska S.A z Poznania za SPYVIEW – System Archiwizacji i Zarządzania Obiegiem Dokumentów. System przeznaczony do tworzenia archiwów o bardzo dużej pojemności.

III stopnia

➲ Śląska Grupa Informatyczna sp. z o.o. z Katowic za MED-NET 3.5 – Pakiet Zintegrowanych Programów Medycznych. Program medyczny do zastosowań w pracy w tzw. „białej sferze" szpitala;

➲ COLUMBUS POLSKA SP. Z O.O., z Warszawy za CONCORDE XAL, kompleksowe, nowoczesne rozwiązanie do zarządzania firmą, zgodne z systemami Unii Europejskiej, uzupełnione o polskie przepisy prawne. Możliwość zastosowania na wielu platformach systemowych.

Wyróżnienia:

➲ X-COM SC z Sosnowca za produkt APTEKA. Program do obsługi Apteki w architekturze klient-serwer z rozpoznawaniem kodów kreskowych i programowym rozliczeniem VAT. Pracuje w środowiskach WIN95 i NT. Może zastąpić w przyszłości rozwiązania na platformie systemowej DOS.

➲ Zakład Usług Informatycznych REKORD Sp. z o.o.. z Bielska-Białej za produkt: MANAGER. Za przygotowanie niedrogiego, krajowego produktu klasy MRP II.

➲ JUNISOFTEX Sp. z o.o. z Gliwic za produkt: PERFECT EXPERT Moduł SPK – NET. System wspomagający zarządzanie przedsiębiorstwem z wykorzystaniem Internetu dla firm rozproszonych z możliwością księgowania w różnych planach kont. Atrakcyjny dla firm posiadających siedziby w różnych krajach.

Nagroda Prezesa Górnośląskiego Oddziału Polskiego Towarzystwa Informatycznego

Wydawnictwa Naukowe PWN z Warszawy za produkt: Komputerowy Słownik Języka Polskiego PWN. Za kulturotwórcze wykorzystanie nowoczesnych technik informatycznych. Słownik języka polskiego, wyrazów obcych oraz wbudowana poprawna wymowa słów trudnych.

Nagroda Prezydenta Miasta Katowice

Konsorcjum 2SI - Sieciowe Systemy Informacyjne, z Katowic za produkt: System dla Ośrodków Pomocy Społecznej SERCE-POMOST. Za podjęcie i realizację niezwykle ważnego społecznie zagadnienia wspomagania pracy ośrodków pomocy społecznej przy wykorzystaniu nowoczesnych technologii, gwarantujących znaczną elastyczność i niezawodność działania.

Nagroda sponsorska PLL LOT

ALBION sp. z o.o. z Wrocławia za produkt: TATRY – Góry Polskie. Za krzewienie turystyki poprzez wykorzystanie środków informatycznych i umożliwienie zapoznania się z folklorem i geografią regionu tatrzańskiego.

Nagroda sponsorska LUFTHANSY

Wydawnictwa Szkolne i Pedagogiczne z Warszawy za produkt: Galeria Malarstwa Polskiego – Muzeum Narodowe w Warszawie. Za wysoki poziom edycyjny i wartościowe narzędzie edukacyjne.

Dane hurtowo

Robert Wrembel

Rozwijające się gałęzie przemysłu, handlu, medycyny, nauki wymagają składowania i przetwarzania ogromnych ilości danych. Dane są zwykle przechowywane w systemach informatycznych posiadających różne struktury i wykorzystujących różne modele danych (np. hierarchiczne, relacyjne, obiektowe), w dokumentach tekstowych czy arkuszach kalkulacyjnych.

Ponieważ przedsiębiorstwa, instytucje, organizacje (producenci danych) są często geograficznie rozproszone, więc same dane mają również charakter rozproszony. Heterogeniczność i rozproszenie informacji utrudnia do nich dostęp i ich analizę. Dlatego do składowania i przetwarzania dużych ilości danych, pochodzących z różnych systemów, stosuje się tzw. **hurtownie (magazyny) danych** (*data warehouses*).

Dane przechowywane w magazynach wykorzystuje się między innymi do analizy trendów, anomalii, poszukiwania reguł w systemach wspomagania decyzji (*Decision Support Systems*) i odkrywania wiedzy (*Data Mining*). W systemach tych baza danych jest eksplorowana za pomocą skomplikowanych zapytań wykorzystujących łączenie, filtrowanie, agregowanie ogromnej liczby danych. Są to tzw. zapytania typu OLAP (*On Line Analythical Processing*).

Innym zastosowaniem hurtowni danych jest tworzenie i przechowywanie kopii bezpieczeństwa danych strategicznych [15].

Ze względu na złożoność zapytań OLAP i ogromną liczbę danych, do których są one adresowane, czas odpowiedzi systemu bazy danych może być bardzo długi, na przykład dziesiątki minut, godziny, dni. Kluczowym problemem jest więc jego efektywność. Dlatego wiele prac koncentruje się na poszukiwaniu nowych struktur danych, bardziej efektywnych algorytmów przetwarzania danych i sposobów ich składowania.

Obecnie wiele czołowych firm w dziedzinie baz danych oferuje serwery wspierające technologię hurtowni danych [3]. Są to:

- *Oracle7*, *Oracle8* i *Oracle Express Server* – Oracle Corporation;
- *DB2* – IBM;
- *Sybase IQ* – Sybase, Inc.;
- *OnLine Dynamic Server*, *OnLine Extended Parallel Server* i *OnLine Workgroup Server* – Informix Software, Inc.;
- *Red Brick Warehouse* – Red Brick Systems, Inc.;
- *Teradata* – NCR, *Adabas C* i *Adabas D* – Software AG;
- *Essbase* – Arbor Software Corporation.

Artykuł ten jest poświęcony omówieniu zagadnień związanych z technologią hurtowni danych.

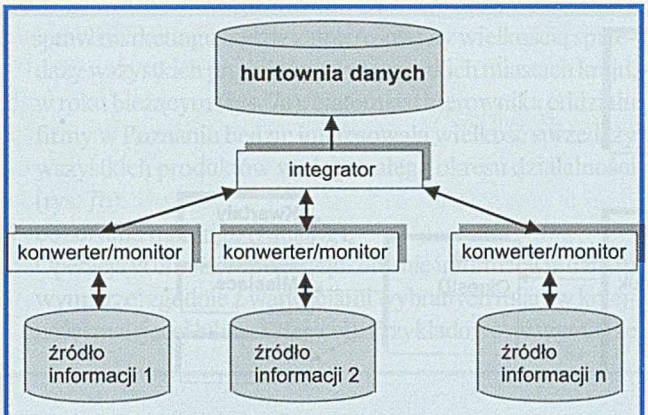
Architektura systemu hurtowni danych

Ważnym wymaganiem systemów informatycznych jest dostęp do informacji pochodzących zarówno z heterogenicznych baz danych (wykonanych w różnych technologiach), jak i ze źródeł innych niż bazy danych, na przykład arkuszy kalkulacyjnych, dokumentów tekstowych, plików ustrukturalizowanych HTML, SGML. Informacje przechowywane w różnych źródłach są integrowane w hurtowni danych. Architektura takiego systemu została przedstawiona na rysunku 1 [17]. Obiekty oznaczone jako *źródło danych* reprezentują wymienione wyżej heterogeniczne źródła informacji. Z każdym z takich źródeł jest związana warstwa oprogramowania – *konwerter/monitor*.

Zadaniem modułu *konwertera* jest transformowanie danych z formatu wykorzystywanego w źródle do formatu wykorzystywanego w hurtowni. Dlatego dla każdego modelu danych źródłowych konieczne jest zastosowanie specyficznego modułu *konwertera*. Przykładowo, jeśli źródło przechowuje dane w dokumentach tekstowych, a hurtownia została zaprojektowana z wykorzystaniem modelu relacyjnego, to *konwerter* musi zapewnić poprawne odwzorowanie danych z plików w struktury modelu relacyjnego.

Zadaniem modułu *monitora* jest wykrywanie zmian w danych źródłowych i ich przekazywanie do warstwy oprogramowania *integratora* (po uprzedniej konwersji do modelu danych hurtowni). Sposób wykrywania zmian w danych źródłowych zależy od własności samych źródeł. Wyróżnia się cztery następujące rodzaje źródeł danych [17]:

- aktywne (*active sources*), tzn. posiadające zaimplementowane mechanizmy wyzwalaczy, które informują *monitor* o zmianach zachodzących w danych źródłowych;
- utrzymujące dzienniki operacji wykonywanych na danych źródłowych (*logged sources*) – zmiany są wykrywane przez analizę zawartości dziennika przez moduł *monitora*;
- umożliwiające wydawanie zapytań (*queryable sources*) – w celu wykrycia zmian w danych źródłowych *monitor* okresowo wydaje zapytania do wszystkich źródeł;
- wspierające mechanizm migawek (*snapshot sources*) – w tego rodzaju źródłach zmiany wykrywa się przez porównanie zawartości kolejnych migawek.



Rys. 1. Architektura systemu hurtowni danych

W wypadku gdy problem wykrywania i propagowania zmian w danych źródłowych jest trudny do rozwiązania, stosuje się okresowe uaktualnianie całej hurtowni w trybie wsadowym. Wówczas hurtownia jest niedostępna dla użytkowników. Ta technika jest wykorzystywana również w wypadku, gdy aktualność danych nie jest konieczna dla poprawnego działania firmy, instytucji czy organizacji oraz gdy dopuszczalna jest czasowa niedostępność danych.

Kolejną warstwę oprogramowania stanowi moduł *integratora*. Jest on odpowiedzialny za uaktualnianie danych w hurtowni. Proces ten składa się nie tylko z wpisywania, uaktualniania i usuwania danych, ale również z wstępnego ich przetwarzania (*data scrubbing*), tj. filtrowania, eliminowania duplikatów, usuwania niespójności, obliczania agregatów, łączenia danych pochodzących z wielu źródeł.

Istniejące komercyjne systemy zarządzania bazami danych wspierające technologię hurtowni danych umożliwiają integrację danych pochodzących z systemów relacyjnych, dostarczanych przez największych wytwórców. Firma Oracle dostarcza oprogramowanie o nazwie *Oracle Loader* i *Oracle Transparen Gateways* umożliwiające dostęp oraz eksportowanie i importowanie danych z większości systemów relacyjnych. IBM umożliwia wczytywanie i transformowanie danych do formatu bazy DB2 za pomocą oprogramowania o nazwie *DataPropagator*, *DataRefresher* i *DataHub*. *SourcePoint* jest oprogramowaniem firmy Software AG służącym do ładowania hurtowni danymi pochodzącymi z innych relacyjnych źródeł. Informix realizuje dostęp do danych zgromadzonych w bazie *Oracle*, *Sybase* i *DB2* za pomocą oprogramowania *Informix–Enterprise Gateway*. Oprogramowanie firmy Sybase umożliwia bezpośrednie wczytywane danych z wielu systemów relacyjnych. Wszystkie z istniejących na rynku systemów wspierają standard ODBC (*Open Database Connecitvity*) dostępu do bazy danych.

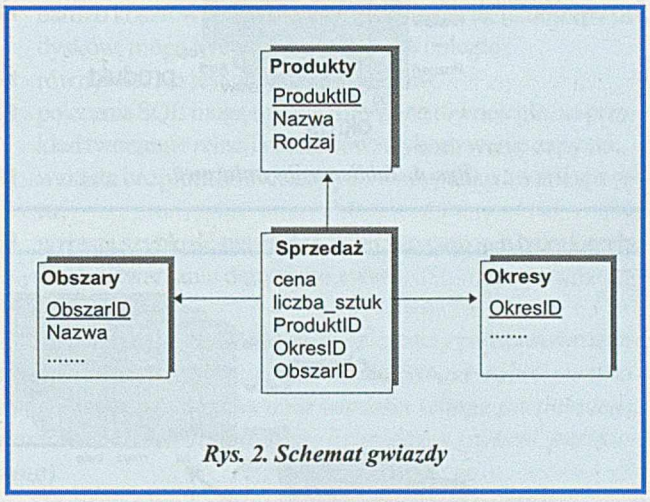
Projektowanie hurtowni danych

Bazy danych wspierające technologię hurtowni danych (technologię OLAP) można podzielić na dwa rodzaje, ze względu na wykorzystywane przez nie modele danych. Pierwszy to bazy *relacyjne*, nazywane również *ROLAP* (*Relational OLAP*), a drugi to bazy *wielowymiarowe* (*multidimensional*), nazywane również *MD–OLAP* (*MultiDimensional OLAP*).

ROLAP

Hurtownia danych tego typu jest zbudowana na podstawie systemu zarządzania relacyjną bazą danych posiadającego mechanizmy efektywnego przetwarzania zapytań typu OLAP (por. z efektywnością systemu hurtowni danych). Zwykle schemat takiej hurtowni posiada strukturę gwiazdy (*star schema*) lub strukturę bardziej złożoną, przypominającą płatek śniegu (*snowflake schema*). W celu skrócenia czasu potrzebnego na wyznaczenie wyników zapytania relacje bazy danych są często denormalizowane, na przykład zawierają wartości zagregowane, są wynikiem połączenia wielu innych relacji [2].

Przykładowy schemat o strukturze gwiazdy przedstawia rysunek 2. Centralna relacja *Sprzedaż* zawiera informacje o sprzedaży pewnych produktów, w pewnych obszarach geograficznych, w określonym czasie. Relacje *Produkty*, *Obszary* i *Okresy* są nazywane *wymiarami* (*dimensions*), natomiast relacja centralna jest nazywana relacją *faktów* (*fact table*). Atrybuty relacji faktów przechowujące informacje o sprzedaży są nazywane *miarami* (*measures*), na przykład *cena*, *liczba_sztuk*. Relacja faktów – *Sprzedaż* zawiera również atrybuty *ProduktID*, *ObszarID*, *OkresID*, których wartości wskazują na odpowiednie wymiary.



Rys. 2. Schemat gwiazdy

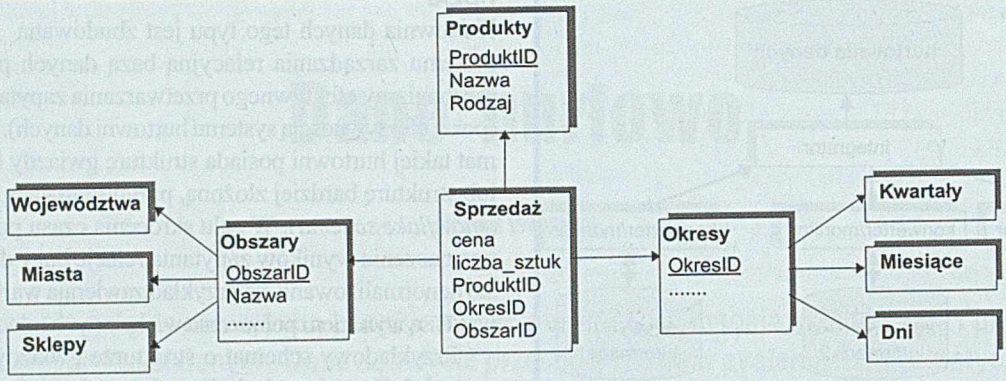
Jeśli wymiary tworzą hierarchie, to schemat hurtowni danych ma często postać płatka śniegu [7]. Przykładowy schemat o takiej strukturze został przedstawiony na rysunku 3.

MD–OLAP

Hurtownia danych zaprojektowana w technologii MD–OLAP do przechowywania danych wykorzystuje wielowymiarowe tablice (*multidimensional arrays, datacubes*). Tablice te zawierają wstępnie przetworzone (m.in. zagregowane) dane pochodzące z wielu źródeł. Przykładowa tablica, zawierająca trzy wymiary: *obszar*, *okres* i *produkt* oraz zagregowane informacje o sprzedaży samochodów w poszczególnych latach, w wybranych miastach, została przedstawiona na rysunku 4.

Analizę danych wielowymiarowych wspomagają specjalne operatory, do których należą:

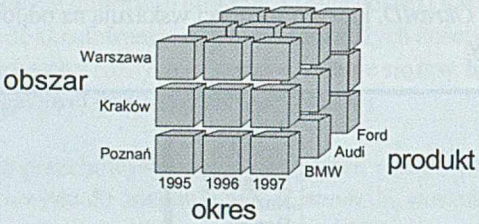
- wyznaczanie punktu centralnego (*pivoting*);
  Operacja ta polega na wskazaniu miary i określeniu wymiarów, w których wybrana miara będzie prezentowana. Przykładowo, w wymiarze produktu reprezentującego samochód marki BMW i wymiarze obszaru reprezentującego sklepy



Rys. 3. Schemat płatka śniegu

województwa poznańskiego może być prezentowana liczba sprzedanych samochodów.

- rozwijanie (*drillng down*);

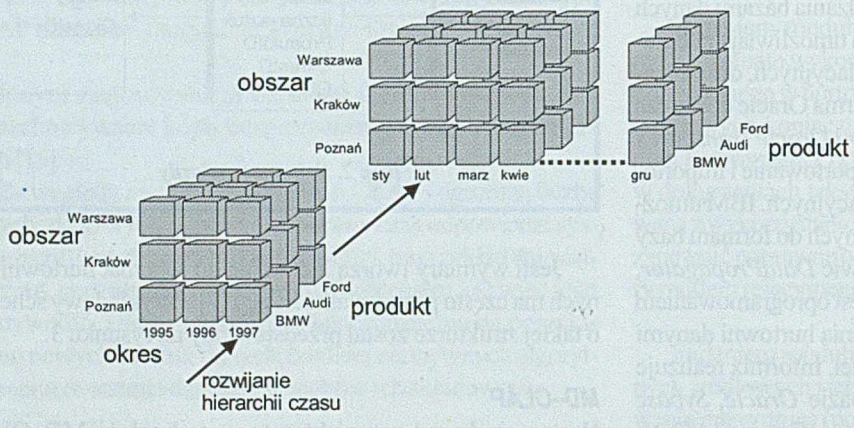


Rys. 4. Tablica trójwymiarowa

Rozwijanie polega na zagłębianiu się w hierarchię danego wymiaru w celu przeprowadzenia bardziej szczegółowej analizy danych. Jako przykład rozważmy informacje o sprzedaży samochodów marek BMW, Audi i Ford, w latach 95, 96 i 97, w poszczególnych miastach (por. rys. 4). W celu dokonania analizy sprzedaży w poszczególnych miesiącach 1997 roku należy rozwinąć hierarchię reprezentującą czas, tj. rok 97 (rys. 5). Analiza sprzedaży w poszczególnych dniach wybranego miesiąca będzie możliwa po rozwinięciu hierarchii reprezentującej ten miesiąc.

- zwijanie (*rolling up*);

Zwijanie jest operacją odwrotną do rozwijania i polega na nawigowaniu w górę hierarchii danego wymiaru. Dzięki tej operacji można przeprowadzać analizę danych zagregowanych na wyższym poziomie hierarchii wymiarów.



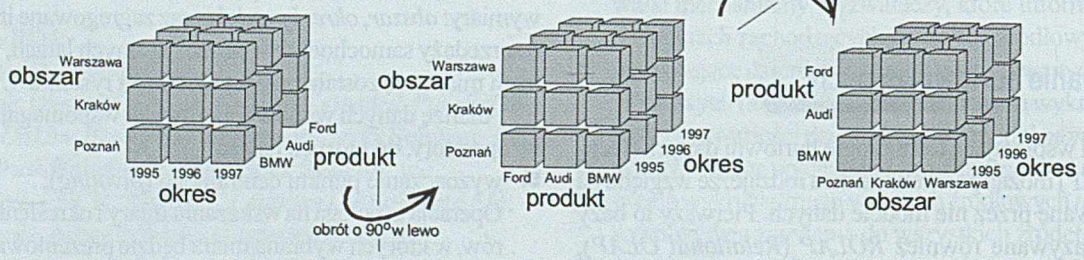
Rys.5. Operacja rozwijania hierarchii wymiaru

- obracanie (*rotating*);

Operacja obracania (rys. 6) umożliwia prezentowanie danych w różnych układach. Celem jej jest zwiększenie czytelności analizowanych informacji.

- wycinanie (*slicing and dicing*);

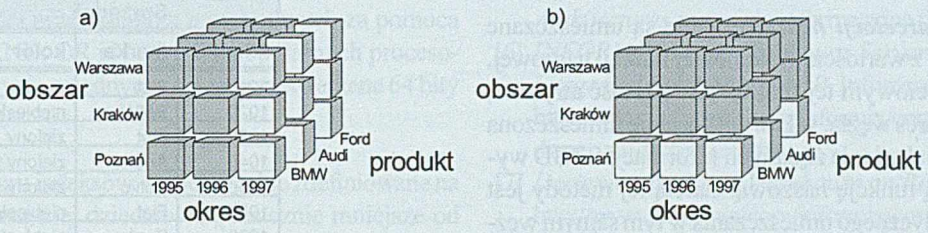
Operacja ta umożliwia zawężenie analizowanych danych do wybranych wymiarów, a w ramach każdego z wymiarów – zawężenie analizy do konkretnych jego wartości. Przykładowo, dyrektor do



Rys.6. Operacja obracania

spraw marketingu będzie zainteresowany wielkością sprzedaży wszystkich produktów, we wszystkich miastach kraju, w roku bieżącym (rys. 7a). Natomiast kierownika oddziału firmy w Poznaniu będzie interesowała wielkość sprzedaży wszystkich produktów w ciągu całego okresu działalności (rys. 7b).

- obliczanie rankingu (*ranking*);
  Operacja ta umożliwia uporządkowanie informacji w danym wymiarze, zgodnie z wartościami wybranych miar (w kolejności malejącej lub narastającej). Przykładowo, w wymiarze roku 1997 można uporządkować marki samochodów zgodnie z narastającym porządkiem liczby sprzedanych egzemplarzy.



Rys. 7. Wycinanie danych w różnych wymiarach

Efektywność systemu hurtowni danych

Aby systemy relacyjnych baz danych mogły efektywnie przetwarzać zapytania typu OLAP i zarządzać danymi rzędu terabajtów (1TB = 1024 GB), muszą posiadać nowe własności zwiększające ich efektywność. Nowymi cechami systemów relacyjnych są między innymi przetwarzanie równoległe, parcelacja danych i nowe techniki indeksowania.

Przetwarzanie równoległe

Przetwarzanie równoległe (*parallel processing*) polega na rozbiciu złożonych operacji na mniejsze, które następnie są wykonywane równolegle, na przykład, na wielu procesorach lub komputerach. W efekcie, czas wykonania całej operacji jest krótszy. W wypadku hurtowni danych najczęściej równolegle przetwarza się zapytania, sortuje dane, wykonuje operacje odczytu i zapisu na dysk, buduje tablice i indeksy oraz wczytuje dane do hurtowni.

Przetwarzanie równoległe wspierają między innymi systemy zarządzania bazami danych: *Oracle7* i *Oracle8* (Oracle Corporation), *DB2* (IBM), *OnLine Extended Parallel Server*, *OnLine Dynamic Server* (Informix), *Red Brick Warehouse* (Red Brick), *Sybase IQ* (Sybase) [3].

Parcelacja danych

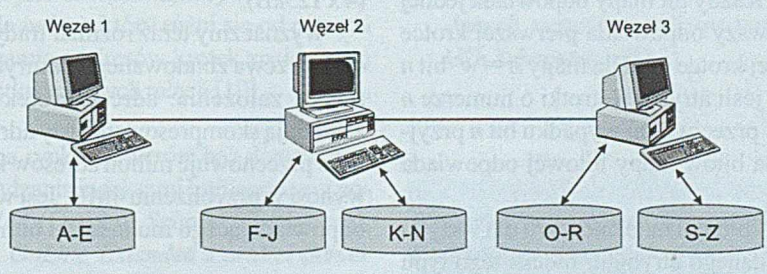
Parcelacja danych (*data partitioning*) umożliwia automatyczne rozpraszanie danych (pochodzących z jednej lub wielu relacji) na wiele dysków, znajdujących się w tym samym lub wielu węzłach (komputerach) sieci. Dzięki podziałowi dużej relacji na mniejsze:

- bardzo kosztowne operacje wejścia/wyjścia, tj. dostępu do dysków, mogą być wykonywane równolegle ;
- równoważone jest obciążenie dysków;
- polecenia SQL mogą być wykonywane równolegle, na przykład tworzenie relacji i indeksów, wykonywanie zapytań;
- wzrasta bezpieczeństwo danych w wypadku awarii sprzętu;
- wzrasta szybkość tworzenia kopii zapasowych bazy i szybkość odtwarzania danych po awarii.

W komercyjnych SZBD stosuje się cztery podstawowe techniki parcelacji danych: *round–robin* (*round–robin partitioning*), parcelacja *bazująca na wartości* (*range partitioning*), *haszowa* (*hash partitioning*), *hybrydowa* (*hybrid partitioning*).

Technika ***round–robin*** umożliwia równomierne rozproszenie danych w węzłach sieci. Przykładowo, jeśli w sieci znajdują się trzy węzły, to pierwsza krotka relacji zostanie umieszczona w węźle pierwszym, druga – w węźle drugim, trzecia krotka – w węźle trzecim, czwarta – znów w węźle pierwszym itp. Rozwiązanie to ma jednak wadę. Ponieważ dane są rozproszone w sposób przypadkowy, więc odnalezienie żądanych informacji wymaga przeszukania wszystkich węzłów.

W wypadku parcelacji ***bazującej na wartości*** rozmieszczenie danych w sieci zależy od wartości samych danych. Przykła-



Rys. 8. Parcelacja danych bazująca na wartości

dowo, relacja zawierająca informacje o klientach sieci supermarketów może być podzielona zgodnie z wartością pierwszej litery nazwiska, jak na rysunku 8 [5]. Węzeł pierwszy posiada jeden dysk, na którym umieszczono klientów o nazwiskach rozpoczynających się od liter A do E. Węzły drugi i trzeci posiadają po dwa dyski, na których umieszczono klientów o nazwiskach odpowiednio z zakresów F–J, K–N, O–R, S–Z.

Ten sposób rozpraszania danych jest efektywny dla zapytań wykorzystujących zakresy wartości w predykatach selekcji, ponieważ umożliwia szybki dostęp do danych z żądanego zakresu, bez potrzeby przeszukiwania wszystkich węzłów.

W wypadku ***parcelacji haszowej*** dane są umieszczane w węzłach zgodnie z wartością systemowej funkcji haszowej. Argumentem wejściowym tej funkcji jest wartość atrybutu, a jej wynikiem – adres węzła, w którym zostanie umieszczona krotka. W celu odnalezienia żądanych informacji SZBD wykorzystuje tę samą funkcję haszową. Zaletą tej metody jest możliwość automatycznego umieszczania w tym samym węźle krotek pochodzących z różnych, powiązanych z sobą relacji. W ten sposób zwiększa się efektywność wykonywania operacji łączenia krotek, gdyż łączone z sobą krotki znajdują się w tym samym węźle.

Parcelacja hybrydowa umożliwia dwustopniowe rozpraszanie danych. W kroku pierwszym dane są umieszczane w poszczególnych węzłach za pomocą parcelacji haszowej. W kroku drugim dane są umieszczane na poszczególnych dyskach danego węzła za pomocą parcelacji bazującej na wartości. Dzięki tej technice wzrasta równomierność rozproszenia danych i obciążenia węzłów.

Indeksowanie danych

Dla bardzo złożonych zapytań typu OLAP, operujących na ogromnej liczbie danych, standardowe indeksy w postaci B–drzew okazują się nieefektywne, ponieważ:

- nie zapewniają wystarczająco szybkiego dostępu do danych;
- ich rozmiar jest zbyt duży, przez co wzrastają koszty ich przetwarzania, przechowywania i utrzymywania.

Dla tej klasy zastosowań stosuje się więc nowe struktury indeksów, tj. *indeksy bitmapowe* (*bit–mapped indexes*) i *indeksy połączeniowe* (*join indexes*) [10] [11] [12][1][18].

Ideą ***indeksów bitmapowych*** jest wykorzystanie pojedynczych bitów do zapamiętania informacji o tym, że dana wartość atrybutu występuje w określonej krotce relacji. Dla każdej unikalnej wartości atrybutu jest przechowywana tablica bitów, zwana *mapą bitową*. Każdy bit mapy odpowiada jednej krotce relacji R – bit pierwszy odpowiada pierwszej krotce relacji R, bit drugi – drugiej krotce itp. Dla mapy A='w' bit n przyjmuje wartość jeden, jeśli atrybut A krotki o numerze n przyjmuje wartość 'w'. W przeciwnym wypadku bit n przyjmuje wartość zero. Liczba bitów mapy bitowej odpowiada liczbie krotek relacji R.

Indeks bitmapowy jest zbiorem map bitowych dla wszystkich unikalnych wartości danego atrybutu. Indeks tego typu może również posiadać strukturę B–drzewa, w którego liściach zamiast adresów rekordów są przechowywane mapy bitowe.

Powyższa koncepcja zostanie zilustrowana przykładem. Tabela 1 przedstawia fragment relacji *Sprzedaż* przechowującej informacje o sprzedaży samochodów. Dla atrybutu *kolor* zdefiniowano indeks bitmapowy składający się z dwóch map bitowych. Pierwsza z nich opisuje te krotki relacji, dla których atrybut *kolor* przyjmuje wartość '*zielony*', a druga – te krotki, dla których *kolor*='*niebieski*'. Ponieważ atrybut ten przyjmuje dwie różne wartości, więc indeks zawiera dwie mapy bitowe. Bit pierwszy mapy *kolor*='*zielony*' przyjmuje wartość 1, co oznacza, że atrybut *kolor* pierwszej krotki przyjmuje wartość '*zielony*'.

Sprzedaż		
klientID	marka	kolor
1010	Fiat	zielony
1020	BMW	niebieski
1030	Fiat	zielony
1040	Audi	zielony
1050	Volvo	zielony
1060	Fiat	niebieski
1070	Ford	niebieski
1080	Opel	zielony
1090	Opel	niebieski
1100	Ford	zielony

Tabela 1. Przykładowa relacja **Sprzedaż**

Natomiast bit drugi mapy przyjmuje wartość 0, ponieważ wartością atrybutu *kolor* krotki drugiej nie jest '*zielony*'.

kolor	
zielony	niebieski
1	0
0	1
1	0
1	0
1	0
0	1
0	1
1	0
0	1
1	0

Tabela 2. Indeks bitmapowy dla atrybutu **kolor**

Podstawową zaletą indeksów bitmapowych jest ich mały rozmiar dla atrybutów o wąskiej dziedzinie wartości. Jako przykład rozważmy relację R zawierającą milion krotek. Przyjmijmy, że na atrybucie A przyjmującym cztery różne wartości utworzono indeks bitmapowy. Indeks ten składa się z czterech map bitowych o rozmiarze miliona bitów każda, co daje rozmiar każdej z map w przybliżeniu 125kB. Łączny rozmiar indeksu bitmapowego wynosi zatem w przybliżeniu 500kB (4 x 125kB).

Wyznaczmy teraz rozmiar tradycyjnego indeksu w postaci B–drzewa zbudowanego na atrybucie A, przyjmując następujące założenia: adresy krotek są czterobajtowe, liście zawierają skompresowane listy adresów krotek. Wówczas indeks przechowuje milion adresów krotek, a więc jego rozmiar wynosi w przybliżeniu 4MB. Jest więc osiem razy większy od odpowiadającego mu indeksu bitmapowego.

Rozważmy jednak tę samą relację R, której atrybut A przyjmuje 64 różne wartości. W tym wypadku indeks bitmapowy

będzie miał rozmiar 8MB (64 x 125kB). Będzie więc większy niż odpowiadający mu indeks w postaci B–drzewa.

W celu zmniejszenia rozmiarów indeksów bitmapowych stosuje się ich automatyczną kompresję. Między innymi w systemach *Oracle7*, *Oracle8* i *Sybase IQ*.

Indeksy bitmapowe są bardziej efektywne od indeksów w postaci B–drzewa tylko dla określonej klasy zapytań kierowanych do bazy danych. Są to zapytania wykorzystujące dużą liczbę predykatów warunkowych oraz zapytania wykorzystujące funkcję COUNT. Większa efektywność tych indeksów wynika z:

- dużej szybkości przetwarzania map bitowych za pomocą operatorów AND, OR i NOT – dla popularnych procesorów 64–bitowych, w jednym cyklu są przetwarzane 64 bity mapy;
- małego rozmiaru indeksów – indeksy takie zdefiniowane na atrybutach o wąskiej dziedzinie są znacznie mniejsze od indeksów w postaci B–drzewa, dzięki czemu mogą być przechowywane w pamięci operacyjnej;
- możliwości wykonywania operacji logicznych i funkcji COUNT bezpośrednio na indeksach bitmapowych (znajdujących się w pamięci operacyjnej), a nie na samych krotkach. W rezultacie, system nie wykonuje kosztownych odczytów danych z dysku w czasie przetwarzania polecenia. Dane są odczytywane dopiero po wykonaniu wszystkich operacji logicznych na indeksach. Ich wynikiem jest mapa bitowa opisującą tylko te krotki, które spełniają warunki selekcji.

Indeksy bitmapowe wykazują jednak mniejszą efektywność dla zapytań wyszukujących dane z zadanych zakresów (operatory >, < itp.), ponieważ dla takiej klasy zapytań należy wykonać wiele operacji sum logicznych na mapach bitowych dla wszystkich wartości z zadanego zakresu.

Innym rodzajem indeksu zwiększającym szybkość przetwarzania danych jest tzw. ***indeks połączeniowy*** (*join index*). Indeks tego typu łączy z sobą krotki z różnych relacji posiadające tę samą wartość atrybutu połączeniowego [10], jest więc strukturą zawierającą zmaterializowane połączenie wielu relacji. Indeks taki posiada strukturę B–drzewa zbudowanego na atrybucie połączeniowym relacji (bądź na wielu takich atrybutach). Liście indeksu zawierają wspólne wartości atrybutu połączeniowego relacji wraz z listami adresów krotek w każdej z łączonych relacji.

Odmianą tego indeksu jest tzw. ***bitmapowy indeks połączeniowy*** (*bit–mapped join index*), który różni się od powyższego tym, że w liściach zamiast adresów krotek znajdują się mapy bitowe opisujące krotki łączonych relacji [10].

Spośród istniejących na rynku serwerów baz danych mechanizmy tworzenia i zarządzania indeksami bitmapowymi posiadają: *Oracle7* i *Oracle8* (Oracle), *Sybase IQ* (Sybase), *OnLine Dynamic Server*, *OnLine Extended Parallel Server* i *OnLine Workgroup Server* (Informix) i *Red Brick Warehouse* (Red Brick Systems).

Literatura

[1] Abbey M., Corey M. J., *Oracle8 – A Beginner's Guide*, Osborne McGraw–Hill, 1997, ISBN 0-07-882393-5.

[2] Colliat G., *OLAP, Relational, and Multidimensional Database Systems*, SIGMOD Record, wolumin 25, nr 3, wrzesień 1996.

[3] Data Management Strategies, Cutter Information Corp., http://www.cutter.com/itgroup/

[4] Edelstein H., *Technology Analysis: Faster Data Warehouses*, http://techweb.cmp.com/iw/556/56olbit/htm

[5] *INFORMIX–OnLine Extended Parallel Server for Loosely Coupled Cluster and Massively Parallel Processing Architectures*, materiały informacyjne firmy Informix, 1997.

[6] *INFORMIX–Data Warehouse Servers. The Evolution of Integrated Relational OLAP*, Informix White Paper, 1996, http://www/informix/com/informix/corpinfo/zines/whitpprs/dwserver/dwserver.htm

[7] *Designing the Data Warehouse on Relational Databases*, http://www/informix/com/informix/corpinfo/zines/whitpprs/stg/metacube.htm

[8] *An Introduction to Multidimensional Database Technology*, http://www.kenan.com/acumate/mddb_toc.htm

[9] O'Neil P., *Model 204 Architecture and Performance*, Springer–Verlag Lecture Notes in Computer Science 359, materiały konferencyjne Second International Workshop on High Performance Transactions Systems, Asilomar, CA, 1987.

[10] O'Neil P., Graefe G., *Multi–Table Joins Through Bitmapped Join Indices*, SIGMOD Record, wolumin 24, nr 3, wrzesień 1995.

[11] O'Neil P., Quass D., *Improved Performance with Variant Indexes*, materiały konferencyjne SIGMOD, Tuscon, Arizona, USA, 1997.

[12] *Bitmapped Indexes in Oracle7*, an Oracle White Paper, 1995.

[13] *A Red Brick Systems White Paper*, http://www.redbrick.com/rbs/whitepapers/datawah_wp.html

[14] Sarawagi S., *Indexing OLAP data*, Data Engineering Bulletin 20(1), 1997.

[15] Silberschantz A., Stonebreaker M., Ullman J., *Database Research: Achievements and Opportunities Into the 21st Century*, raport NFS Workshop on the Future of Database Systems Research, 1995.

[16] *Optimizing Interactive Performace for the Data Warehouse*, http://www.novasys.com.cy/sysprod06.thm

[17] Widom J., *Research Problems in Data Warehousing*, materiały konferencyjne, 4–th International Conference on Information and Knowledge Management (DIKM), 1995.

[18] Wrembel R., *Nowe struktury indeksów dla magazynów danych*, materiały konferencyjne III Jesiennej Szkoły PLOUG, Zakopane, 1997.

Robert Wrembel jest pracownikiem Instytutu Informatyki Politechniki Poznańskiej
e–mail: wrembel@cs.put.poznan.pl

Inspekcje oprogramowania

Janusz Górski

W ramach cyklu życia oprogramowanie istnieje w wielu różnych reprezentacjach. Jedną z nich jest wykonywalny kod programu. Inspekcje dokumentacji to jedna z najbardziej skutecznych metod wykrywania defektów w dokumentach. Inspekcje mogą być stosowane do dowolnej dokumentacji. Dzięki temu stanowią jedną z podstawowych technik projakościowych, stosowanych we wczesnych etapach rozwoju oprogramowania, wtedy gdy kod programu jeszcze nie istnieje. Są więc istotnym uzupełnieniem testowania, które jest tradycyjnie wykorzystywane w stosunku do kodu programu. Artykuł przedstawia proces inspekcji oraz korzyści płynące ze stosowania tej techniki.

Koszt naprawy defektu wprowadzonego na etapie analizy wymagań, zidentyfikowanego i usuniętego w trakcie projektowania systemu jest 10-krotnie większy niż koszt jego usunięcia na etapie wymagań. Usunięcie takiego błędu w trakcie implementacji systemu może kosztować już 100 razy więcej, a w trakcie eksploatacji - nawet 1000 razy (jest to tzw. reguła 1:10 [1]). Zapewnienie jakości jest ukierunkowane na zapobieganie, wykrywanie i usuwanie defektów we wszystkich dokumentach tworzonych w ramach cyklu życia oprogramowania. Odgrywa zasadniczą rolę w procesie wytwarzania oprogramowania i powinno towarzyszyć jego każdej fazie i aktywności. Wyniki analiz dotyczących jakości oprogramowania wskazują, że większość (około 60-70%) defektów zostaje wprowadzona w fazach analizy i projektowania, a jedynie 30-40% - w trakcie kodowania. Jeśli uwzględnimy, że testowanie umożliwia wykrycie jedynie około 50% defektów (i to w większości powstałych w procesie kodowania), okazuje się, że ograniczenie się wyłącznie do testowania nie jest skuteczną drogą zapewniania jakości oprogramowania. Należy wprowadzić do użytkowania metody, które umożliwiają eliminację defektów w możliwie najkrótszym czasie po ich wprowadzeniu.

Cele inspekcji

Jeden z najbardziej skutecznych sposobów rozwiązania powyższego problemu stanowią inspekcje dokumentacji. Jest to metoda zespołowego wykrywania defektów w produktach procesu wytwarzania oprogramowania. Ze względu na jej wysoką efektywność i uniwersalność jest to jedna z najważniejszych metod zapewnienia jakości.

Rozumiane w sensie potocznym inspekcje (przeglądy, recenzje) dokumentów stanowią szeroko i od dawna stosowaną technikę pracy z dokumentacją, polegającą na tym, że (inna niż autor) osoba dokonuje krytycznego przeglądu dokumentu, poszukując zawartych w nim defektów. W postaci sformalizowanej oraz w stosunku do oprogramowania metoda inspekcji została zaproponowana przez M. E. Fagana w latach siedemdziesiątych. Metoda ta przyniosła nadspodziewanie dobre wyniki w zakresie poprawy zarówno jakości produktu, jak i efektywności procesu jego wytwarzania.

Pośród celów inspekcji wymienić można:

- wykrywanie i identyfikację defektów;
- usunięcie tych defektów;
- formalne potwierdzenie jakości produktu;
- doskonalenie procesu wytwarzania oprogramowania na podstawie wyników analizy przyczyn znalezionych defektów;
- doskonalenie samego procesu inspekcji.

Proces inspekcji

Inspekcja obejmuje wiele kroków, prowadzonych na podstawie zdefiniowanych procedur i dokumentów pomocniczych, z udziałem zespołu o jawnie przydzielonych rolach. Każdy krok ma zdefiniowane cele, warunki wejściowe, wyjściowe, wykorzystywane dokumenty, produkty, procedury postępowania oraz uczestników.

Proces inspekcji zostaje zainicjowany w chwili zgłoszenia przez właściciela dokumentu (np. autora) zapotrzebowania na przeprowadzenie inspekcji (*faza rozpoczęcia*). Zgłoszenie to trafia do odpowiedniej komórki, która wyznacza kompetentnego lidera inspekcji. Lider sprawdza, czy produkt oraz produkty źródłowe (czyli te, na podstawie których powstał dokument przewidziany do inspekcji) spełniają określone kryteria wejściowe. Celem tych działań (*faza wejścia*) jest niedopuszczenie do inspekcji produktów źle przygotowanych, których analiza byłaby jedynie stratą czasu. *Faza planowania* obejmuje prace przygotowawcze, np. ustalenie liczby spotkań kontrolnych, wyznaczenie uczestników, przygotowanie harmonogramu prac itp. *Spotkanie inicjujące* ma na celu wyjaśnienie wszystkim uczestnikom inspekcji jej celów, harmonogramu, oczekiwań w stosunku do nich i ich pracy, rozdanie materiałów, przydział ról, wyjaśnienie procedur inspekcji, określenie parametrów wydajnościowych (np. tempa kontroli indywidualnej). *Kontrola indywidualna* dokumentu jest wykonywana osobno przez

każdego z kontrolerów na podstawie produktów źródłowych, procedur kontroli, obowiązujących zasad i list kontrolnych. Głównym celem pracy kontrolera jest znalezienie jak największej liczby defektów w badanym dokumencie. *Spotkanie kontrolne* jest kluczową fazą procesu inspekcji. W tej fazie są zbierane i dokumentowane wyniki indywidualnej pracy kontrolerów. Wykryte defekty podlegają klasyfikacji i są przedstawiane w ramach wspólnej struktury danych. Dodatkowym celem spotkania kontrolnego jest znajdowanie i dokumentowanie nowych defektów i zagadnień, które nie zostały wykryte w trakcie pracy indywidualnej. W trakcie spotkania są również zgłaszane i dokumentowane sugestie poprawy samego procesu inspekcji, list kontrolnych, procedur i reguł. Adresatem tych zagadnień jest komórka odpowiedzialna za inspekcje. W ten sposób w trakcie spotkania kontrolnego są pozyskiwane efekty pracy grupowej. Kolejnym krokiem inspekcji jest *edycja*, w ramach której są rozwiązane problemy udokumentowane podczas spotkań kontrolnych. Krok ten najczęściej realizuje autor ocenianego dokumentu. W kolejnym kroku *kontynuacji* lider inspekcji dokonuje sprawdzenia wyników fazy edycji, rozsyła odpowiednim osobom sugestie zmian produktów źródłowych oraz propozycje poprawy procesu inspekcji. Kolejnym krokiem jest *wyjście*, będące formalnym zatwierdzeniem przez lidera wyników inspekcji. Po zakończeniu tego etapu dokument zostaje oficjalnie udostępniony do dalszego wykorzystania, z podaniem szacunkowej liczby pozostałych w nim defektów.

Dodatkowym etapem inspekcji może być tzw. *burza mózgów*, zwoływana w celu podjęcia próby identyfikacji przyczyn powstania najpoważniejszych defektów wykrytych w badanym dokumencie. Zwykle prowadzi to do sugestii poprawy procesu wytwarzania oprogramowania.

Role uczestników

W inspekcji uczestniczy zespół osób pełniących różne role. Możliwe jest spełnianie więcej niż jednej roli przez tę samą osobę. Poniżej krótko omówiono poszczególne role.

Lider inspekcji jest postacią centralną. Od niego w dużej mierze zależy jej jakość i efektywność. Zakres odpowiedzialności lidera obejmuje kontrolę spełnienia warunków wejścia dokumentu, organizację i przeprowadzenie spotkania inicjującego, wybór kontrolerów (w porozumieniu z autorem), przygotowanie i przeprowadzenie spotkania kontrolnego, klasyfikację i dokumentowanie wykrytych defektów i zagadnień, przekazanie raportu z inspekcji działowi zapewnienia jakości dla celów statystycznych, kontrolę spełnienia warunków wyjścia dokumentu. Lider jest odpowiedzialny za sprawne i skuteczne zrealizowanie całego procesu.

Autor aktywnie uczestniczy w większości kroków inspekcji. To on podejmuje decyzję o przedłożeniu swojego dokumentu do inspekcji. Bierze udział w spotkaniu inicjującym i najczęściej prowadzi również kontrolę indywidualną swojego produktu oraz uczestniczy w spotkaniach kontrolnych. Autor pełni również kluczową rolę podczas fazy edycji dokumentu i jest odpowiedzialny za dokonanie odpowiednich zmian.

Sekretarz. Jego zadaniem jest dokumentowanie zagadnień, zgłaszanych w trakcie spotkania kontrolnego. Rolę tę może pełnić każdy członek zespołu.

Kontroler. Zwykle w zespole uczestniczy kilku kontrolerów, których zadaniem jest znalezienie jak największej liczby defektów oraz ich czytelne udokumentowanie. Zakres uwagi różnych kontrolerów może być rozdzielony na przykład poprzez wykorzystanie różnych list kontrolnych.

Dokumentacja

Proces inspekcji jest wspierany odpowiednią dokumentacją obejmującą: *przewodnik inspekcji* zawierający definicję wzorcowego dla danej instytucji procesu inspekcji w ujęciu ogólnym oraz, z punktu widzenia poszczególnych ról, *zasady i listy kontrolne* wykorzystywane w trakcie kontroli indywidualnej, jak również różne *formularze* związane z poszczególnymi krokami inspekcji (np. formularz zgłoszenia potrzeby inspekcji, formularz planu inspekcji, formularz listy zagadnień/defektów wykrytych podczas inspekcji, formularz podsumowania inspekcji itp.).

Efektywność inspekcji

Efektywność inspekcji należy rozumieć w kategoriach krótko- i długoterminowych. W pierwszym wypadku chodzi o wpływ inspekcji na jakość badanego dokumentu w odniesieniu do poniesionych nakładów. W drugim wypadku chodzi o wpływ procesu inspekcji na poprawę praktyki wytwarzania oprogramowania w danej instytucji (łącznie z samym procesem inspekcji). O ile pierwszy z powyższych celów może być osiągnięty stosunkowo łatwo, poprzez zebranie i interpretację danych odnoszących się do indywidualnej inspekcji, to w drugim wypadku niezbędne jest podjęcie dodatkowego wysiłku zbierania, dokumentowania i analizowania dużej ilości informacji niezwiązanych bezpośrednio z badanym dokumentem. Są to informacje o charakterze jakościowym, na przykład przyczyny pojawienia się różnych rodzajów defektów, oraz ilościowym, na przykład liczba znalezionych głównych zagadnień na jednostkę czasu kontroli indywidualnej, tempo dokumentowania zagadnień podczas spotkania kontrolnego, pracochłonność całego procesu. Zbieranie tych danych musi być odpowiednio zaplanowane oraz włączone w sferę zarządzania projektem.

Inspekcje nie są darmowe. Wymagają znacznego nakładu prac organizacyjnych i merytorycznych. Pracochłonność pojedynczej inspekcji może łatwo przekroczyć sto roboczogodzin. Dodatkowo należy uwzględnić koszty szkoleń, zarządzania, zbierania danych, analizy metryk itp. Z drugiej strony, wartość inspekcji wyraża się liczbą wykrytych defektów w stosunku do tych, które nie zostały wykryte. Dopiero całościowa ocena uwzględniająca wpływ inspekcji na cały proces wytworzenia produktu informatycznego umożliwia wyciągnięcie właściwych wniosków. Publikowane doświadczenia jednoznacznie oceniają inspekcję jako przedsięwzięcie bardzo opłacalne. Poza zyskiem czysto ekonomicznym, związanym z redukcją całkowitej pracochłonności, są wskazywane inne korzyści ze stosowania tej techniki. Dzięki stosowaniu inspekcji we wszystkich fazach wytwarzania defekty są usuwane równomiernie, w miarę ich wprowadzania do kolejnych reprezentacji produktu, a nie jedynie w trakcie testowania. Systematyczne stosowanie inspekcji

oraz dokumentowanie ich wyników wspomaga doskonalenie procesu wytwarzania poprzez ujawnianie tych jego obszarów, które należy poprawić w pierwszej kolejności. Dodatkowe korzyści wynikają z podnoszenia jakości i kwalifikacji wytwórców oprogramowania poprzez zapoznawanie się z wynikami prac innych osób (podczas kontroli indywidualnej), udział w szkoleniach, spotkaniach kontrolnych i „burzach mózgów".

T. Gilb [3] podaje wiele danych ilościowych charakteryzujących wpływ inspekcji na proces wytwarzania oprogramowania: wzrost produktywności o 30% - 100%, skrócenie czasu wytwarzania o 10% - 30%, 5-10-krotna redukcja czasu i kosztu testowania, redukcja kosztów pielęgnacji o rząd wielkości, niemal „automatyczna" poprawa jakości procesu i produktu, eliminacja poślizgów w dostarczaniu produktów.

Podsumowanie

Inspekcje stanowią jedną z technik wykorzystywanych w ramach systemu zapewnienia jakości, który wyznacza właściwy kontekst do jej zastosowania. Muszą służyć osiąganiu jawnie zdefiniowanych celów, a ich zastosowanie musi podlegać normalnym procesom planowania i zarządzania. Sukces w zastosowaniu inspekcji zależy od wielu czynników. Jednym z najważniejszych jest dobre osadzenie procesu inspekcji w ramach systemu zapewniania jakości oraz ścisłe zdefiniowanie kroków, ról, zadań i zasad składających się na ten proces. Wprowadzanie inspekcji do praktyki danej organizacji powinno odbywać się według opracowanego wcześniej scenariusza, uwzględniającego takie czynniki, jak:

- pełne poparcie wyższych szczebli zarządzania;
- świadomość kadry inżynieryjno-technicznej;
- szkolenia;
- tworzenie bazy metryk;
- dokumentowanie.

Szczególnie ważne są czynniki socjologiczne, psychologiczne, relacje interpersonalne i ogólna atmosfera w organizacji. Inspekcja może łatwo przerodzić się w ocenę autora, a nie wytworzonego przez niego dokumentu. Prowadzi to bezpośrednio do konfliktów na gruncie personalnym i stanowi jedno z najpoważniejszych zagrożeń dla skuteczności inspekcji. Podobny efekt dają próby wykorzystania przez kierownictwo metryk zbieranych w trakcie inspekcji do oceny uczestniczących w niej ludzi, a nie do oceny i poprawy samego procesu.

Literatura

[1] B. W. Boehm, *Software Engineering Economics*, Prentice-Hall, 1981.

[2] M. E. Fagan, *Design and Code Inspections to Reduce Errors in Program Development*, IBM systems Journal, 15(3).

[3] T. Gilb, D. Graham, *Software Inspection*, Addison-Wesley Publishing Company, 1993.

[4] D. P. Freedman, G.M. Weinberg, *handbook of Walkthroughs, Inspections and Technical Reviews,* Dorset House Pub., New York, 1990.

[5] M. Jurkowlaniec, B. Runge, *Inspekcje produktów informatycznych*, Informatyka 1997, nr 11.

Prof. **Janusz Górski**
Politechnika Gdańska, Katedra Zastosowań Informatyki
e-mail: jango@pg.gda.pl

infosystem '99

MIĘDZYNARODOWE TARGI ELEKTRONIKI, TELEKOMUNIKACJI I TECHNIKI KOMPUTEROWEJ

13 – 16.04.1999

Termin zgłoszeń: 15.12.1998

Zakres branżowy targów:

- sprzęt komputerowy
- oprogramowanie
- sieci komputerowe
- automatyka przemysłowa
- technika biurowa
- telekomunikacja
- usługi, szkolenia, doradztwo
- aparatura kontrolno-pomiarowa
- urządzenia technologiczne dla przemysłu elektronicznego
- elementy i podzespoły elektroniczne

Rabat 20% dla firm, które nadeślą zgłoszenia udziału do 15.12.1998

Promocyjna oferta dla zgłaszających się na INFOSYSTEM po raz pierwszy

Międzynarodowe Targi Poznańskie Sp. z o.o.
Zespół Organizacji Targów B2
Głogowska 14
60-734 Poznań
Tel. 061/869 25 99, 866 43 14
Fax 061/866 07 07
http://www.mtp.pol.pl

Na początku roku, w bardzo rzeczowym artykule [1], wypomniano polskim informatykom i ich zleceniodawcom siedem grzechów popełnianych przy budowaniu dużych systemów informacyjnych dla gospodarki i administracji publicznej:

- **Podejmowanie decyzji skomputeryzowania systemu zanim zaistnialo określające jego ramy ustawodawstwo;**
- **Błędne określenie potrzeb;**
- **Chciwość;**
- **Źle podpisane kontrakty;**
- **Złe i niestabilne prawo;**
- **Uleganie tradycjom Polski resortowej;**
- **Za dużo polityki.**

Zaminowany teren dużych systemów komputerowych

Jacek Kryt

Sądzę, że warto podzielić się z Czytelnikami swoją wiedzą na ten temat, ale w odniesieniu do Ameryki Północnej.

Obok uprawiania początków informatyki, w połowie lat pięćdziesiątych zaczęto tu automatyzować istniejące od lat systemy płac, księgowe i organizacyjne w dużych przedsiębiorstwach, ale dopiero w 1968 rok uporządkowano uniwersyteckie studia informatyki (ACM *Computer Science Curriculum*). Do tego czasu programistami potrzebnymi pionierom komputeryzacji, słusznie z reszta, zostali kwalifikujący się do tej pracy byli użytkownicy systemu ręcznego, którzy go doskonale znali z doświadczenia. Po przeszkoleniu na kilkudniowym kursie programowania szybko oprogramowali to, co należy. Wywiązali się z zadania dobrze i przez wiele lat, uzupełniając w pracy swą wiedzę, byli kręgosłupem kadry analityków i programistów. Niestety nie na zawsze. Dyplom z informatyki wkrótce stał się w wielu firmach warunkiem otrzymania pracy obsłudze i modernizacji ich systemów informatycznych (SI).

40 lat to dość czasu, aby zrobić wiele głupstw. Uniknęliśmy jednak niektórych dziur wspomnianych w [1]. Inaczej niż w Polsce, nie zmienialiśmy systemu gospodarki równolegle z systemami informatycznymi, a w rezultacie w USA stare, ręczne i „dotarte" systemy, po małych poprawkach, nadawały się do adaptacji.

Ustawy czy decyzje ramowe są normalnie uzupełniane procedurami. Prawo u nas ewoluuje z tym, że administracja państwowa uprzedza o planowanych zmianach przepisów przed czasem, ułatwiając uwzględnienie tych zmian w oprogramowaniu. Podejście resortowe spotyka się czasem, ale w zakresie ochrony dóbr osobistych obywatela. Natomiast źle podpisanych kontraktów jest teraz niewiele. Mamy za to bardzo liczne grupy nacisku broniące swych, nie zawsze powszechnie rozumianych, interesów związanych z wymuszaniem komputeryzacji. Chciwość nie jest tylko polskim wynalazkiem.

Niekompletne lub błędne określenia potrzeb to częsta przyczyna tego, że systemy nie zadawalają użytkownika, który za nie płaci. To chyba uniwersalny problem nierozwiązany od czasu, gdy programiści, byli użytkownicy, zakończyli swą działalność w SI.

Trudno wiarygodnie oszacować, jaki jest procent systemów, które okazały się „niewypałami". Cytowane czasem wartości według administracji amerykańskiej (70-80 %), nie są miarodajne, gdyż dotyczą systemów, które były podejrzane o złą realizację i dlatego dokładniej audytowane. Znane są systemy, które działają tak, jak życzą sobie tego użytkownicy, którzy są z nich zadowoleni, a ponadto przynoszą oczekiwane wymierne korzyści gospodarcze. Jednak jest wiele niepowodzeń, wynikających z różnych przyczyn, o których wygodniej jest nie myśleć, gdyż nieda się tych przyczyn usunąć wyłącznie przez wymiany kodu maszynowego.

Psychologia informatyka i użytkownika

Używanie komputera do przetwarzania tekstu czy innych prostych operacji nawet początkującemu użytkownikowi daje poczucie siły i iluzję bezwzględnej władzy nad maszyną. U informatyków to poczucie jest szczególnie mocne, gdyż to oni piszą szczegółowe instrukcje nakazujące maszynie, co ma robić.

„Niektórzy specjaliści z dziedziny przetwarzania danych (a także część kadry kierowniczej) uważają, że wszystkie funkcje wykonywane przez system należy zaprogramować w taki sposób, aby nic nie pozostawiać człowiekowi - operatorowi systemu. Rolę człowieka ogranicza programista, który twierdzi, że wszystko można przedstawić w postaci programu. Podobnie postępują niefachowcy, będący niekiedy nadal pod wrażeniem „mózgu elektronicznego", sprzedawcy maszyn, a także ich nabywcy, którzy sądzą, że jeśli tak wiele płacą za maszynę, to powinna ona wszystko robić" [2].

Informatycy modelują skomplikowane procesy gospodarcze czy społeczne, analizując, abstrahując i...upraszczając je, często nie do poznania. Tam gdzie sami, jako odpowiedzialni

analitycy, nie udokumentowali wyczerpująco wymagań użytkownika albo nie byli w stanie ich zrozumieć, zamiast uzupełniać swoją wiedzę, arbitralnie robią własne założenia, decydując co ważne, a co nie, co użytkownik powinien wiedzieć i używać, a co powinno być z góry odrzucane. Informatycy wierzą niezbicie w swój racjonalny i ilościowy sposób interpretacji rzeczywistości. Unikalne doświadczenie użytkownika, jego intuicja, zdrowy rozsadek, hierarchia wartości i wiedza rzadko są brane przez programistów pod uwagę przy budowie systemu.

Nie uwzględniają również tego, że warunki na rynku użytkownika zmieniają się nieraz radykalnie i często, że użytkownik może mieć różne skłonności do akcji czy podejmowania ryzyka, mieć słabsze lub mocniejsze nerwy, problemy osobiste lub zdrowotne, które wszystkie razem wpłyną na jego zainteresowanie i interpretację danych biznesowych. Od 40 lat informatycy zbywają te ważne czynniki powiedzeniem, że użytkownik nie wie, czego chce.

Podkreśla się tylko ciągle wagę coraz nowszej technologii. Panuje niezachwiana wiara, że ta następna, lepsza technologia, szybsze maszyny i programy narzędziowe lub bardziej przyjazne dla użytkownika interfejsy ekranowe, rozwiążą wszystkie problemy.

Pomieszanie pojęć – dane i informacja

Co gorsze, informatycy są nadal przekonywani na uczelniach, i wielu naukowców w to wierzy, że to, co ostatecznie podadzą użytkownikowi po przetworzeniu danych przez komputer, będzie dla użytkownika cenną informacją. To kształtuje ich mentalność i postawę oraz umacnia wiarę w pierwszoplanowe znaczenie technologii informatycznej. Użytkownik musi być zadowolony z tego, co dostanie od informatyków.

Niestety, rzeczywistość jest inna. Cztery dekady doświadczeń wykazują, że buduje się często systemy, które nie realizują tych żądań, jakich wymagał użytkownik. Produkują dane, które go nie interesują.

Powszechność wspomnianych powyżej zjawisk sugeruje, że muszą one mieć jakąś wspólną przyczynę. Moim zdaniem tkwi ona w pomieszaniu pojęć „dane" i „informacja." Te dwa terminy, prawie powszechnie używa się jako równoważne. Niektórzy używają ich na przemian, inni w jednym zdaniu używają ich razem .

Gdy dziesięć tysięcy lat temu, w erze przekazu ustnego (*oral tradition*), danych było mało, to cokolwiek się przekazywało w rozmowie, było bardzo istotne. Jeśli coś nie interesowało odbiorcy, to momentalnie okazywał to rozmówcy. W rezultacie wszystkie **dane**, przekazywane w obu kierunkach, stawały się jednocześnie **informacją** i wiedzą obu uczestników rozmowy.

Te czasy są już dawno za nami. Dziś utożsamianie informacji z danymi, może technicznie wygodne, jest anachronizmem. Radykalna zmiana technologii komunikowania się, zwłaszcza od czasów Gutenberga, (książki, gazety, telegramy, faksy i Internet) spowodowała, że potop przekazywanych danych zwiększył się o wiele rzędów wielkości. Zalani danymi musimy wiele z nich odrzucać, gdyż nasze instrumenty dostrzegania i analizowania są zbyt wolne, by podołać temu nawałowi. Broniąc się, sklecamy różne mentalne zapory, klasyfikacje i filtry elektroniczne. Robimy wszystko, co możemy, by do naszych umysłów dotarły tylko te dane, które nas najbardziej interesują.

Stąd ..."Informacją dla człowieka są dane dostrzeżone albo zauważone, wybrane i zorganizowane przez ich odbiorcę, ze względu na jego subiektywne ludzkie zainteresowania, wywodzące się z jego instynktów, uczuć, doświadczeń, intuicji, zdrowego rozsądku, wartości, wierzeń, osobistej wiedzy czy mądrości. Jednocześnie przetwarzane przez jego poznawcze i mentalne procesy i bez śladu wcielone w jego zapamiętaną wiedzę" [2]. Wszystko inne to tylko bezużyteczne dane.

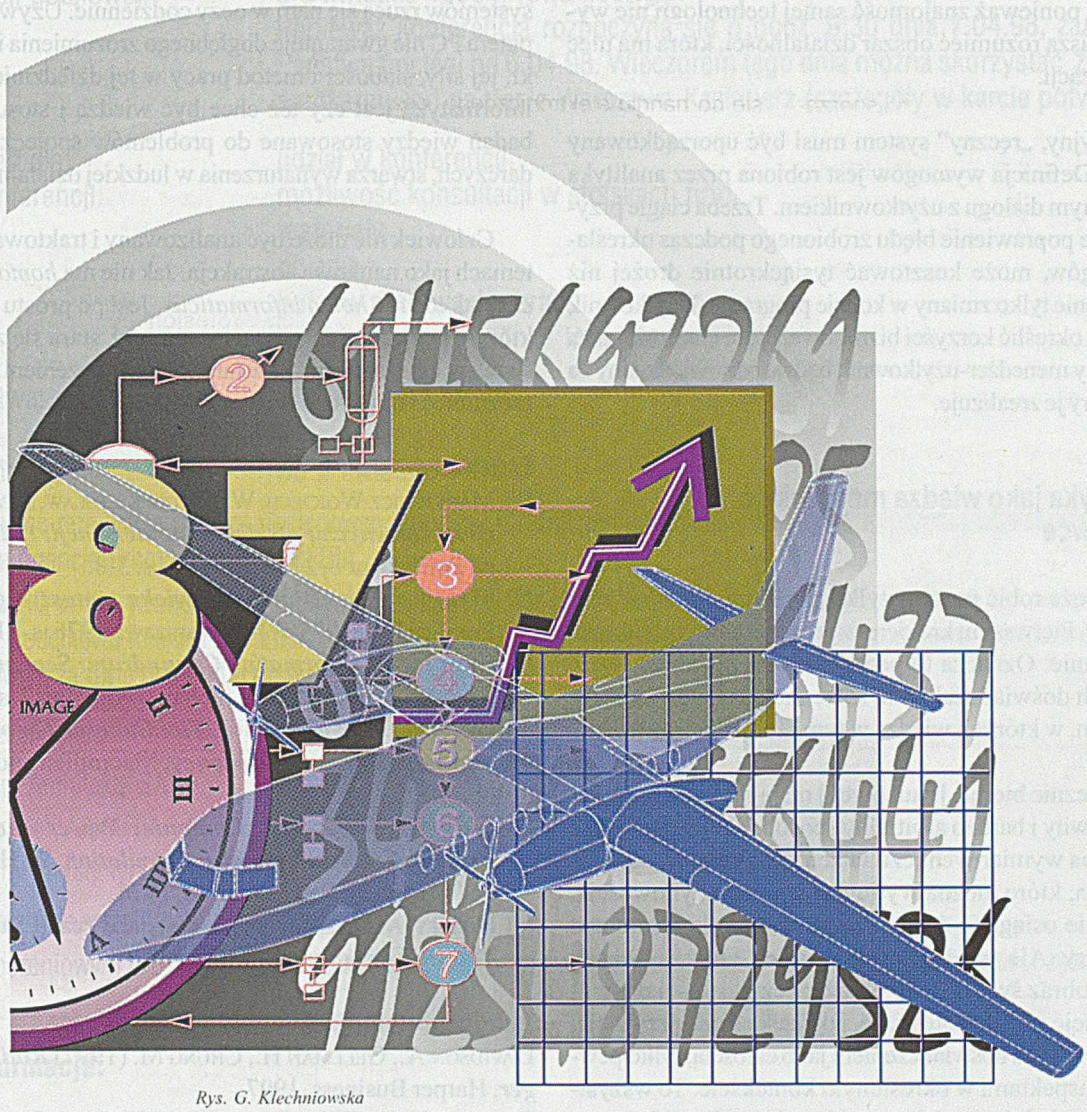
Złożoność i zmienność systemu komputeryzacji

Systemy przedsiębiorstw czy administracji państwowej są wielkimi, złożonymi i „żyjącymi" systemami społecznymi i gospodarczymi. Znajomość wielu dyscyplin naukowych jest potrzebna, by je dogłębnie zrozumieć. A nawet wtedy trudno podjąć decyzję, ponieważ mało jest faktów, prawidłowości i niezbitych dowodów, które pozwolą przewidzieć, jaki osiągnie się wynik w danych okolicznościach. Bywa, że tylko z powodu wielkości projektów trudno ogarnąć ich całość. W miarę postępu budowy wychodzą na jaw różne przeoczone pierwotnie szczegóły. Ponieważ jest tyle niewiadomych w tych rozważaniach, obszar takich systemów pozwala na swobodę, jakiej nie ma się w naukach ścisłych. W naukach społecznych i gospodarczych każdy może pozować na autorytet i tak dobrać sobie elementy swego rozumowania, by konkluzja była taka, jaką sobie wymarzył. Im mniej się zna na tych sprawach, tym szybciej dojdzie do wniosków, które mu odpowiadają. Nie grozi mu, jak mówią Anglosasi, „paralysis by analysis". Może ufnie obwieścić, że doskonale rozumie systemy społeczno-gospodarcze i ludzi, którzy się nimi zajmują..., gdyż naprawdę tak mu się wydaje. Informatycy, na różnych szczeblach odpowiedzialności, nie zdają sobie sprawy, że nie są wolni od takich złudzeń.

Nie jest to odwzajemnione. Nieznajomość specyficznego żargonu nauki X uniemożliwia, laikom i niewtajemniczonym naukowcom z innych dziedzin nauki, zrozumienie naukowca z tego obszaru. Daje mu to poczucie wyższości i często niezasłużony autorytet specjalisty od tajemnej wiedzy, która w praktyce, przy małym wysiłku, może się okazać łatwa do zrozumienia.

Kyenes, ekonomista [3], który zarządzając finansami uniwersytetu Oxford, zarobił dla niego fortunę, twierdził, że „...musimy uwzględniać nerwy, histerie i nawet trawienie oraz reakcje na pogodę tych, od których spontanicznej aktywno-

ści zależą te [inwestycyjne] prognozy... Ludzkie decyzje dotyczące przyszłości, czy to osobiste, czy polityczne, czy ekonomiczne, nie mogą zależeć od ścisłych matematycznych szacunków (*expectations*). Ponieważ nie ma podstaw do robienia takich kalkulacji, to nasz wewnętrzny pęd do aktywności nadaje tok wypadkom, nasza racjonalna część (*self*) wybiera tak, jak można najlepiej, kalkulując gdzie się da, ale często szukając ostatecznego motywu w kaprysie (*whim*), uczuciu czy przypadku."



Rys. G. Klechniowska

Musimy też pamiętać, że zmiana jest jedyną stałą tendencją na rynku globalnym, do którego nieustannie dopasowują się dotknięte jej skutkami przedsięwzięcia. Szybkość dostosowywania się jest warunkiem istnienia na rynku. Informatycy lubią nazywać się inicjatorami zmiany (*agents of change*). Zmieniają maszyny, ikony i klawisze funkcyjne. W trakcie budowy systemu, nieraz arbitralnie, „ulepszają" procesy i rutynę pracy użytkownika. Mniej są na ogół zainteresowani zmianą języka programowania. A już zupełnie nie entuzjazmują ich nieuniknione zmiany zachodzące w kontekście pracy użytkownika wystawionego na nieprzewidywalne przemiany rynku, które wymagają poprawienia napisanych programów. Wprowadzanie i kontrola tych zmian we właściwym czasie jest bardzo istotnym zadaniem w każdym przedsięwzięciu.

Kompleksowość i zmienność systemów komputerowych wskazuje na konieczność rozwijania metodologii i technologii, które ułatwiłyby modernizację programów i wdrażanie zmian szybko, przy małym nakładzie pracy.

Obycie z PC to zbyt mała wiedza jak na decydenta

W Ameryce komputeryzacja zaczęła się około 1950 r., na długo przed pojawieniem się Octów. W krajach, do których komputeryzacja dotarła jednocześnie z PC, wszystko przebiegało inaczej. Zaczęło się od łatwego nabycia PC i jego oprogramowania do przetwarzania tekstu. Kupowano je za małe pieniądze i jeszcze tego samego dnia, jeśli nie odpowiadało, wymieniano na inne. Wszystko odbywało się bez konieczności budowania systemu od podstaw przez odpowiednio wyszkolonych fachowców. Istnieje niebezpieczeństwo, że te osobiste doświadczenia formują sposób postrzegania przez decydentów wszystkich systemów, włącznie

z dużymi wymagającymi wielu lat pracy analityków i programistów.

Przy tworzeniu dużych systemów kolejność działań musi być dobrze przemyślana, by uniknąć fatalnych błędów. Jest istotne, by analitycy mieli odpowiednio dużą wiedzę i przygotowanie, ponieważ znajomość samej technologii nie wystarcza. Muszą rozumieć obszar działalności, która ma ulec komputeryzacji.

Tradycyjny, „ręczny" system musi być uporządkowany i dotarty. „Definicja wymogów jest robiona przez analityka w nieustannym dialogu z użytkownikiem. Trzeba ciągle przypominać, że poprawienie błędu zrobionego podczas określania wymogów, może kosztować tysiąckrotnie drożej niż wprowadzenie tylko zmiany w kodzie programu. Użytkownik musi z góry określić korzyści biznesowe, które chce osiągnąć. Wyznaczony menedżer-użytkownik będzie odpowiedzialny za projekt, który je zrealizuje.

Informatyka jako wiedza może wieść na manowce

„Wiedza może robić postępy tylko przez upraszczanie rzeczywistości. Pierwszym krokiem w procesie upraszczania jest abstrahowanie. Oznacza to pozostawianie poza nawiasem tych danych doświadczalnych, które nie pasują do koncepcyjnych ram, w których wiedza w tym momencie pracuje.

„Praktycznie biorąc, [naukowcy] mają rację, że postępują w ten dziwny i bardzo arbitralny sposób. Skupiając uwagę wyłącznie na wymiernych rezultatach takich elementów doświadczenia, które można wytłumaczyć przyczynowością, byli w stanie osiągnąć dużą i ciągle rosnącą kontrolę nad siłami natury. Ale władza to nie to samo co zrozumienie i naukowy obraz świata, jako obraz rzeczywistości nie wystarcza. Dzieje się tak dlatego, że, jak nauka sama przyznaje, nie zajmuje się ona doświadczeniem jako całością, tylko pewnymi jego aspektami w określonym kontekście. To wszystko jasno rozumieją naukowcy skłonni do filozoficznego myślenia. Niestety niektórzy naukowcy, wielu techników i większość konsumentów gadżetów nie ma ani czasu, ani chęci, by dochodzić do fundamentów i uzasadnień nauki. W rezultacie skłonni są do akceptowania obrazu świata zamkniętego w teoriach naukowych jako całkowity i wyczerpujący obraz rzeczywistości. Skłonni są również przyjmować za dogmat te aspekty doświadczenia, których naukowcy nie biorą pod uwagę, gdyż są niekompetentni, by się nimi zajmować, uważając je za mniej realne aniżeli te aspekty, które wiedza arbitralnie wyabstrahowała z nieskończenie bogatej całości danych faktów." Reszta przy innej okazji.

Konkluzje

Metoda robienia rachunku sumienia pozwoliła poruszyć wiele złożonych spraw. Jedno twierdzenie prowadzi do kilku innych. Starałem się ograniczyć do naszych grzechów. Psychologia informatyków jest specyficznym rezultatem „obcowania" z komputerem. Pomieszanie pojęć „dane" i „informacja" to rezultat trwającej od stuleci historycznej ewolucji środków komunikacji międzyludzkiej. Od czasu masowej produkcji danych utożsamianie znaczenia tych terminów jest błędnym anachronizmem. Złożoność oraz zmienność życia i jego systemów rzuca się nam w oczy codziennie. Używanie komputera PC nie gwarantuje dogłębnego zrozumienia informatyki, jej siły, słabości i metod pracy w tej dziedzinie. Fakt, że informatyka jest czy też chce być wiedzą i stosuje zasady badań wiedzy stosowane do problemów społeczno-gospodarczych, stwarza wynaturzenia w ludzkiej działalności.

Człowiek nie może być analizowany i traktowany w systemach jako naukowa abstrakcja. Jak nie ma *homo economicus*, tak nie ma *homo informaticus*. Jest po prostu człowiek, który wyposażony w pewne przymioty, stara się zrozumieć świat, w którym chce żyć w harmonii z otoczeniem. Trzeba to zrozumieć i mu to ułatwić.

Literatura

[1] Markiewicz Wojciech, Wawrzonek Lesław, ***Siedem Grzechów Głównych Polskiej Komputeryzacji: Dziury w Sieci.*** Polityka nr 8, 21 luty 1998 r., s. 3-8.

[2] Martin James, ***Dialog człowieka z maszyną cyfrową.*** Prentice Hall 1973/WNT, Warszawa 1976, s. 313.

[3] Kryt Jacek, ***Information Conundrum: Semantics...with a payoff, Informatica.*** (Ljubljana) vol. 21 nr 2, s. 305-319.

[4] Keyenes, John M. (1936), ***The General Theory of Employment Interest and Money***, Macmillan and Co, London 1942.

[5] Weizenbaum, Joseph, ***Computer Power and Human Reason. From Judgment to Calculation***. W. H. Freeman and Company, San Francisco, 1976.

[6] Huxley, Aldous, Science, Liberty and Peace, Harper, New York w Weizenbaum, 1946, s.128.

Inne, uzupełniające:

Davidson A., Gellman H., Chung M. (1997) ***Riding the Tiger***, Harper Business, 1997.

Dreyfus, H. L., ***What Computers Still Can't Do***. A critique of artificial reason. The MIT Press, Cambridge, Mass. revised .ed., 1992.

Kent, W., ***Data and Reality, Basic Assumption in Data Processing Reconsiderd***, North-Holland Publishing Co. Amsterdam, 1978.

Ngwenyama, O. K & Lee A. S., ***Critical Social Theory in Electronic Mail***, MIS Quarterly, volume 21, n. 2 s. 145-167, June 1997.

Postman Neil, ***Technopoly, The Surrender of Culture to Technology***, Alfred A. Knopf, New York, 1992.

Jacek Kryt, były profesor Reyson Polytechnic University Toronto, Kanada
e-mail: jkryt@acs.ryerson.ca

INFORMACJE ORGANIZACYJNE

dla uczestników konferencji BAZY DANYCH '98

Termin konferencji:	7-8 kwietnia 1998 r.
Lokalizacja konferencji:	Kazimierz n.Wisłą, Dom Architekta SARP
Termin rezerwacji hotelowej:	6-8 kwietnia 1998 r. (dwa noclegi) Ponieważ konferencja rozpoczyna się o godz. 9:30 dnia 7.04.98, zarezerwowaliśmy dla Państwa noclegi od 6.04.98. Wieczorem tego dnia można skorzystać z naszego połączenia autokarowego na trasie Warszawa-Kazimierz (szczegóły w karcie potwierdzenia udziału).
Zakres świadczeń dla uczestników konferencji:	udział w konferencji, możliwość konsultacji w stoiskach firm, materiały, obiady, bufet w przerwach obrad, koktail i koncert w dn. 7 kwietnia br.
Koszt uczestnictwa:	700 zł
Termin zgłoszeń:	do 2 kwietnia 1998 r.

Osoby zainteresowane rezerwacją miejsca noclegowego prosimy o zaznaczenie rezerwacji hotelu na zgłoszeniu uczestnictwa. Obsługę rezerwacji hotelowej realizuje Agencja Promocji INTER-MEDIA.

Dom Architekta SARP, w którym będzie się odbywać konferencja ma zaledwie 50 miejsc hotelowych - dlatego też dla uczestników konferencji zarezerwowaliśmy dodatkowo miejsca w Ośrodku Arkadia i Domu Turysty. W/w miejsca są zróżnicowane cenowo. Poniższe ceny dotyczą miejsca w pokoju dwuosobowym za dwie doby od 6 do 8.04.98:

nazwa hotelu	koszt dwóch noclegów: 6-8.04.1998
Dom Architekta SARP	170 zł.
Ośrodek ARKADIA	110 zł.
Dom Turysty PTTK	od 130 zł do 350 zł.

Do powyższej kwoty należy doliczyć 22% podatek VAT.

W formularzu zgłoszeniowym prosimy o wpisanie wybranej opcji - miejsca w SARP-ie będą przydzielane w kolejności zgłoszeń.

Dodatkowe informacje:

Centrum Promocji Informatyki Sp. z o.o.
Pani Iwona Nowosielska
tel.: (022) 870-6910, 870-6978
fax: (022) 870-6941, 870-6995

Koordynatorem merytorycznym konferencji jest:
Pani Bożena Prokopowicz
tel./fax: (022) 651-5560, tel.: 0601 362-835
e mail: prokop@pol.pl

Biuro organizacyjne:

Centrum Promocji Informatyki

04-041 Warszawa
ul. Międzyborska 50
tel.: (022) 870-6910, 870-6978
fax: (022) 870-6941, 870-6995
e-mail: cpi@cpi.com.pl

CENTRUM PROMOCJI INFORMATYKI

Zapraszamy na konferencję połączoną z programem konsultacji technicznych i marketingowych

BAZY DANYCH '98

7-8 kwietnia 1998 r., Kazimierz Dolny

Dzień 0 (6 kwietnia 1998 r.)

17:00 **Rozpoczęcie rejestracji uczestników**

18:00 **Konsultacje firmowe dla uczestników.**

Dzień I (7 kwietnia 1998 r.)

TRENDY ROZWOJOWE BAZ DANYCH

9:30 **Dlaczego bazy danych? - prof. M.Muraszkiewicz.**
W wykładzie tym zostaną przypomniane i uporządkowane kwestie terminologiczne i definicyjne dotyczące systemów baz danych. Mocno podkreślone zostaną tu związki pomiędzy systemami i sieciami informacyjnymi (Internet, intranet, extranet). Następnym tematem będą modele danych (relacyjny, sieciowy, obiektowy) oraz rodzaje baz danych (bibliograficzne, faktograficzne, do zarządzania, do produkcji, itd.), a także multimedialność w kontekście systemów informacyjnych. Będą poruszone najważniejsze kwestie dotyczące komunikacji użytkownika z systemami baz danych, a więc głównie języki wyszukiwawcze. Zasygnalizowane zostaną również problemy z zakresu zarządzania, które występują w trakcie projektowania, realizacji i wdrażania systemów baz danych na styku zleceniodawca - wykonawca (sformułowanie założeń, przygotowanie oferty, ocena ofert, negocjacje, dokumentacja fazy implementacji, itd.) Wykład zakończy odpowiedź na pytanie postawione w jego tytule, między innymi przez krótką analizę hipotetycznej sytuacji, w której w przedsiębiorstwie czy organizacji fatalnej awarii uległ system informacyjny.

10:30 **Tendencje w rozwoju i zastosowaniach systemów baz danych - prof. H.Rybiński.**
Omówione zostaną dotychczasowe osiągnięcia w dziedzinie baz danych, a następnie pokazane będą podstawowe trendy, związane np. z bazami obiektowymi, multimedialnością, heterogenicznymi systemami rozproszonymi itd. Przedstawione zostaną przykłady nowych zastosowań, stanowiące prawdziwe wyzwanie dla technologii baz danych.

11:30 **Hurtownie danych - dr T.Traczyk.**
Wykład przedstawia jeden z "gorących tematów" współczesnej informatyki. Opisane zostaną zadania, które spowodowały powstanie i rozwój hurtowni danych. Przedstawiona będzie definicja hurtowni i typowa architektura systemu informacyjnego z hurtownią danych. Wyróżniona zostanie rola centralnej hurtowni danych i wydziałowych składnic danych. Zaprezentowane będą problemy analizy wielowymiarowej i typowe dla niej struktury danych. Omówione będą trudności realizacji przetwarzania analitycznego (OLAP - On-line Analitical Processing) i dwa najczęściej spotykane rozwiązania: relacyjne i wielowymiarowe. Zasygnalizowane zostaną zagadnienia budowy systemu z hurtownią danych i wymagania stawiane używanym do tego narzędziom. Wykład stanowi wprowadzenie do wystąpień dostawców systemów zarządzania bazami danych.

12:30 **Przerwa.**

13:00 **Computer Associates: OpenIngres, Jasmine - możliwości i perspektywy rozwoju.**
Prezentacja firmy Rodan -System

14:00 **Rodzina Sybase Adaptive Server - zoptymalizowane przetwarzanie danych dla różnych typów aplikacji.**
Prezentacja firmy Sysoft

15:00 **Przerwa obiadowa.**

16:00 **Progress - wizja rozwiązań dla sieci lokalnych i rozproszonych.**
Prezentacja firmy CSBI SA

17:00 **Oracle8 baza dla przetwarzania sieciowego.**
Prezentacja firmy Oracle Polska

18:00 **DB2 - niezawodne rozwiązanie dla biznesu dzisiaj i jutro.**
Prezentacja firmy IBM Polska

19:30 **Koncert muzyki organowej.**

20:30 **Koktail na statku płynącym po Wiśle.**

Dzień II (8 kwietnia 1998 r.)

9:00 **Konsultacje firmowe dla uczestników.**

10:30 **Microsoft SQL Serwer - teraźniejszość (6.5) i przyszłość (7.0).**
Prezentacja firmy Microsoft

11:30 **Internetowa implementacja hurtowni danych.**
Prezentacja firmy Informix Software

12:30 **Przerwa.**

13:00 **Rynek systemów zarządzania bazami danych - Andrzej Dyżewski (DiS).**
Można określić bardzo wiele parametrów charakteryzujących popularność różnych systemów zarządzania bazami danych. Może to być liczba programistów piszących aplikacje dla każdego systemu, popularność wśród grup użytkowników, roczna liczba przeprowadzonych transakcji, wielkość sprzedaży, liczba pracujących serwerów, łączna wielkość obsługiwanych baz danych. Każdy z tych parametrów jest ciekawy do analiz, choć nie zawsze prosto postawione pytania znajdują równie łatwe do uzyskania odpowiedzi. Prelegent przedstawi wyniki prac, które pozwalają ocenić wartość niektórych parametrów obecności systemów baz danych na rynku.

13:30 **Dyskusja panelowa dot. uwarunkowań rozwoju zastosowań baz danych w Polsce.**

15:00 **Obiad i zakończenie konferencji.**

Celem konferencji jest prezentacja aktualnego stanu i trendów rozwoju technologii baz danych oraz przedstawienie oferty wiodących w tym zakresie firm informatycznych.

Dokładna nazwa i adres Firmy:

NIP:

Zgłoszenie udziału w konferencji BAZY DANYCH '98 (7-8.04.1998 r.)

Zgłaszamy udział w w/w konferencji następujących osób

imię i nazwisko _____ tel. _____ fax. _____ e-mail _____

imię i nazwisko _____ tel. _____ fax. _____ e-mail _____

Prosimy o rezerwację noclegu (2 doby) w hotelu na warunkach określonych w folderze:

SARP* ☐ ARKADIA* ☐ DOM TURYSTY* ☐ **Przy wybranym hotelu prosimy postawić krzyżyk*

Zobowiązujemy się do wpłacenia należności z tytułu niniejszego zgłoszenia na podstawie faktury VAT i upoważniamy Cl
do wystawienia faktury bez naszego podpisu.

POLSKIE TOWARZYSTWO INFORMATYCZNE

BIULETYN

NUMER 9 (158) ROK XVII PAŹDZIERNIK 1998

50 LAT POLSKIEJ INFORMATYKI 1948-1998

PIOTR FUGLEWICZ WICEPREZYDENTEM CEPIS-u

Trzydziestego sierpnia 1998 roku, podczas odbywającego się w trakcie XV Światowego Kongresu Informatyki IFIP w Wiedniu posiedzenia Rady CEPIS (patrz poprzedni numer Biuletynu oraz http://www.cepis.org/), Piotr Fuglewicz został wybrany na wiceprzewodniczącego (*Vice-President Professional*) Rady Europejskich Profesjonalnych Stowarzyszeń Informatycznych (CEPIS – *Community of European Professional Informatics Societies*). Rada powierzyła nowemu wiceprzewodniczącemu zakres obowiązków obejmujący sprawy rozwoju i promocji pojęcia profesjonalizmu informatycznego w skali europejskiej.

Piotr Fuglewicz (44 l.) jest absolwentem Politechniki Śląskiej w Gliwicach. Ojciec dwóch córek. Właściciel firmy TiP Sp. z o. o., która dostarcza programy do korekty ortograficznej i gramatycznej tekstów dla wszystkich popularnych na rynku narzędzi biurowych (Microsoft, Lotus, WordPerfect). Był prezesem Polskiego Towarzystwa Informatycznego w latach 1993-1996, od 1996 roku pełni funkcję wiceprezesa. W latach 1990-1994 przedstawiciel Polski w Grupie Roboczej ds. Europejskiej Specjalizacji Zawodowej w Informatyce (PDQF TF – *Professional Development and Qualification Task Force*). Od 1993 roku członek Rady (*Board*) CEPIS. Członek Akademii Inżynierskiej w Polsce, współautor książki *CASE dla ludzi*, doradca Zarządu Computerland SA.

W trakcie tego samego zebrania Rada CEPIS przyjęła do swego składu towarzystwa informatyczne z Estonii, Słowacji i Czech, zwiększając tym samym liczbę krajów, które reprezentowane są przez towarzystwa profesjonalnych informatyków, do 26.

Informacja prasowa Oddziału Górnośląskiego PTI

MRĄGOWO '98

Polskie Towarzystwo Informatyczne

oraz

***Initiative for Software Process Improvement – Régions Extérieures* (INSPIRE)**

zapraszają w dniach 16-20 listopada 1998 roku na Czternaste Jesienne Spotkania PTI Mrągowo'98. Tematem konferencji będą *Problemy jakości i niezawodności środków informatyki*. Oprócz wykładów, które wygłoszą wybitni praktycy oraz teoretycy z Polski i zza granicy, oczekujemy prezentacji czołowych producentów środków informatyki, jak również firm doradczych.

Używając porównania z innej dziedziny, można nazwać konferencję mrągowską pokazem kolekcji: wiosna-lato w naszej branży.

Częścią konferencji będzie sesja szkoleniowa projektu INSPIRE. Współpraca PTI z INSPIRE rozpoczęła się w 1998 roku w trakcie konferencji *Doskonalenie procesów konstruowania i używania oprogramowania a integracja systemów informacyjnych*, która odbyła się w Szczyrku. Temat jakości i niezawodności środków informatyki jest, zdaniem PTI, kluczowym problemem na obecnym etapie rozwoju tej dziedziny. Od uczestniczących firm oczekujemy nie tyle solennych naukowych rozważań, ile raczej konkretnych przykładów z życia. O ile konferencja w Szczyrku była konferencją raczej teoretyczną, to zamiarem Mrągowa jest pokazanie, w jaki sposób praktykę wdrażają się w życie (lub wręcz tworzą) czołowi producenci. Poniżej zamieszczamy zgłoszone do tej pory wykłady.

Prof. Piotr Dembiński, *Formalne metody w projektowaniu oprogramowania: mity i rzeczywistość*,
Bogusław Jackowski, Tutorial: *Znaki na niebie, ziemi i gdzie indziej*,
Prof. Henryk Krawczyk, *Modele i ocena jakości systemów informatycznych*,
Borys Stokalski, *Zasady organizacji audytu przedsięwzięć informatycznych*,
Dr Tomasz Traczyk, *Bazy danych, narzędzia dostępu do danych i hurtownie danych*,
Prof. Jan Zabrodzki, *Grafika komputerowa i wirtualna rzeczywistość – stan dzisiejszy*,
Dr Janusz Zalewski, *Systemy czasu rzeczywistego – projektowanie i aspekty praktyczne*.

Bliższych informacji udziela Oddział Górnośląski PTI, tel./fax (32)- 253-96-78

Piotr Fuglewicz

SEES'98
Software Engineering Education Symposium
Poznań, 18-20 listopada 1998r.

http://www.cs.put.poznan.pl/~sees98/

W piątym tegorocznym numerze Biuletynu informowaliśmy Czytelników o pierwszej organizowanej w Polsce międzynarodowej konferencji poświęconej edukacji w dziedzinie inżynierii oprogramowania, która odbędzie się w Poznaniu. Organizatorami Konferencji są: Instytut Informatyki Politechniki Poznańskiej, Komitet Informatyki Polskiej Akademii Nauk oraz Polskie Towarzystwo Informatyczne. Dziś parę dalszych szczegółów.

Swoją obecność na Konferencji potwierdzili goście honorowi zaproszeni do wygłoszenia referatów na kluczowe dla dziedziny tematy:

- **Prof. Nancy Mead, Carnegie-Mellon University** (wtorek, 18.11.1998, godz. 10.35);
- **Prof. David L. Parnas, McMaster University** (środa, 19.11.1998, godz. 10.35);
- **Prof. Ian Sommerville, Lancaster University** (czwartek, 20.11.1998, godz. 10.35).

Ponadto referaty wygłoszą:

- **Cliona McGowan, European Software Institute, Bilbao, Hiszpania;**
- **Aaron Agarwal,Motorola, Schaumburg, USA;**
- **Christophe Debou, Q-Labs Software Engineering GmbH, Niemcy.**

W trakcie regularnych sesji referaty wygłoszą przedstawiciele środowisk uniwersyteckich z jedenastu krajów europejskich i Stanów Zjednoczonych. Odbędą się również dwie sesje plakatowe.

Konferencji towarzyszyć będą:

- 4th Workshop of the Thematic Network on Computing (EU), wtorek, 17.11.1998;
- 3rd IEEE Real-Time Systems Education Workshop, sobota, 21.11.1998.

Dzięki sponsoringowi następujących firm:

- EAPLS (*European Association for Programming Languages and Systems*);
- MOTOROLA;
- INSPIRE - *Initiative for Software Process Improvement*;
- Q-Labs,

organizatorzy uzyskali możliwość zredukowania wysokości opłaty konferencyjnej dla polskich uczestników do następujących kwot:

- 200DM lub 375 zł do 10 października 1998
- 250 DM lub 460zl po 10 października 1998.

Opłata obejmuje uczestnictwo w sesjach, materiały konferencyjne, napoje w czasie przerw, obiady oraz uczestnictwo w przyjęciu powitalnym i uroczystej kolacji konferencyjnej.

Szczegóły dotyczące zgłoszenia można znaleźć pod adresem: http://www.cs.put.poznan.pl/~sees98/

Zapraszamy!

Jubileuszowa Górska Szkoła PTI

W dniach 22-26 czerwca br. odbyła się coroczna, tym razem już Dziesiąta Jubileuszowa Górska Szkoła PTI. Ośrodek „Zagroń" w Szczyrku gościł 145 uczestników konferencji, których tak liczny udział tłumaczyć można frapującym tematem, który brzmiał „Doskonalenie procesów konstruowania i używania oprogramowania a integracja systemów informacyjnych". Organizatorem, jak co roku, był Oddział Górnośląski PTI, zaś współorganizatorami: Sekcja Inżynierii Oprogramowania PTI, *Initiative for Software Process Improvement - Regions Exterieures* (INSPIRE) oraz Inicjatywa Poprawy Procesów Programowania (IP^3). Wykłady wygłaszali przedstawiciele ośrodków naukowych oraz firm komercyjnych, przy czym ci drudzy koncentrowali się bardziej na rozwiązaniach, bowiem komercji w czystej postaci, opierającej się na prezentacji produktów, było niewiele, co chyba należy uznać za dobry wyznacznik kształtujących się tendencji. Zauważył to zresztą Szanowny Jubilat - Leszek Zawistowski, który po raz dziesiąty zaszczycił Górską Szkołę. Z tej też okazji otrzymał z rąk Prezesa OG PTI Jerzego Nowaka dyplom okolicznościowy. Konferencję niewątpliwie uświetnił swą obecnością Prezes Fundacji ECDL Dudley Dolan z Irlandii.

Niestety nie wywiązał się ze swego zobowiązania Jacek Kaczmarski, odwołując wcześniej umówiony występ. Być może nie była to aż tak wielka strata w kontekście umiejętności muzycznych i wokalnych, zaprezentowanych przez niektórych uczestników konferencji podczas wieczornych biesiad - jak widać informatycy posiadają także inne talenty.

Niesprzyjająca, gdyż deszczowa aura nie była w stanie zepsuć dobrego nastroju uczestników. Był czas na dyskusje w kuluarach, nawiązywanie nowych znajomości oraz spacery podczas przerw między opadami. Szczególnie silnie motywowani koledzy zdobywali w rekordowym czasie Klimczok lub Skrzyczne. Dla niezainteresowanych górskimi wędrówkami pozostawał do dyspozycji kryty basen lub sauna.

Czas X Szkoły minął szybko i ciekawie - dzięki dobrej atmosferze i interesującym tematom wykładów, które, mimo iż dotyczyły istotnych i trudnych zagadnień, podane były w przystępnej, a jednocześnie profesjonalnej formie. Dowcipne i rzeczowe przewodniczenie zajęciom osobą Piotra Fuglewicza czynione, potęgowało dobry nastrój oraz integrację wykładowców i słuchaczy.

Piotr Kowalski

Od Biuletynu: Wracając do tematyki logotypu PTI, zamieszczamy piękną, krajobrazową propozycję znaku naszego Towarzystwa. Takie logo znalazło się na okładce materiałów Szkoły. Autorką jest Kol. Mira Minich, której dziękujemy za jego udostępnienie.

Redaktor: EWA ŁUKASIK
e-mail: lukasik@put.poznan.pl, tel. (0-61) 878 23 73
Instytut Informatyki, Politechnika Poznańska,
ul. Piotrowo 3a, 60-965 Poznań

Polskie Towarzystwo Informatyczne, Zarząd Główny
tel. (22) 624 60 61 w.328, tel./fax (22) 652 32 59
URL: http://www.pol.pl/pti
ul. Żelazna 87, 00-879 Warszawa

Jak wiele można przekazać mówiąc tak mało:

www.*moja_firma*.pol.pl

Wykorzystując rozwiązania oferowane przez **Polska OnLine** Wasza firma może:

✓ **Ograniczyć koszty materiałów marketigowych** wykorzystując Internet do publikacji kompletnej oferty towarów i usług, katalogu wyrobów, szczegółowych opisów technicznych i cenników.

✓ **Zredukować koszty komunikacji** (telefony, faksy, poczta) wykorzystując pocztę elektroniczną i elektroniczne listy dystrybucyjne.

✓ **Podnieść prestiż firmy** umacniając jej wizerunek jako firmy nowoczesnej, wykorzystującej Internet do kontaktu z Partnerami na całym świecie.

✓ **Zwiększyć efektywność działań marketingowych** wykorzystując elektroniczne ankiety, budując politykę bezpośredniego kontaktu z Klientem.

✓ **Zmniejszyć koszty operacyjne** wprowadzając system elektronicznych zamówień i informacji dla kontrahentów, wykorzystujący rozwiązania typu Extranet.

✓ **Zwiększyć wydajność firmy** poprzez zastosowanie nowoczesnych systemów dystrybucji i kontroli obiegu informacji, opartych na technologii Intranet i wykorzystujących rozwiązania typu GroupWare i Workflow.

Polska OnLine

- 2 lata najpopularniejszego polskiego WWW
- 6 lat praktyki w projektowaniu systemów dla firm

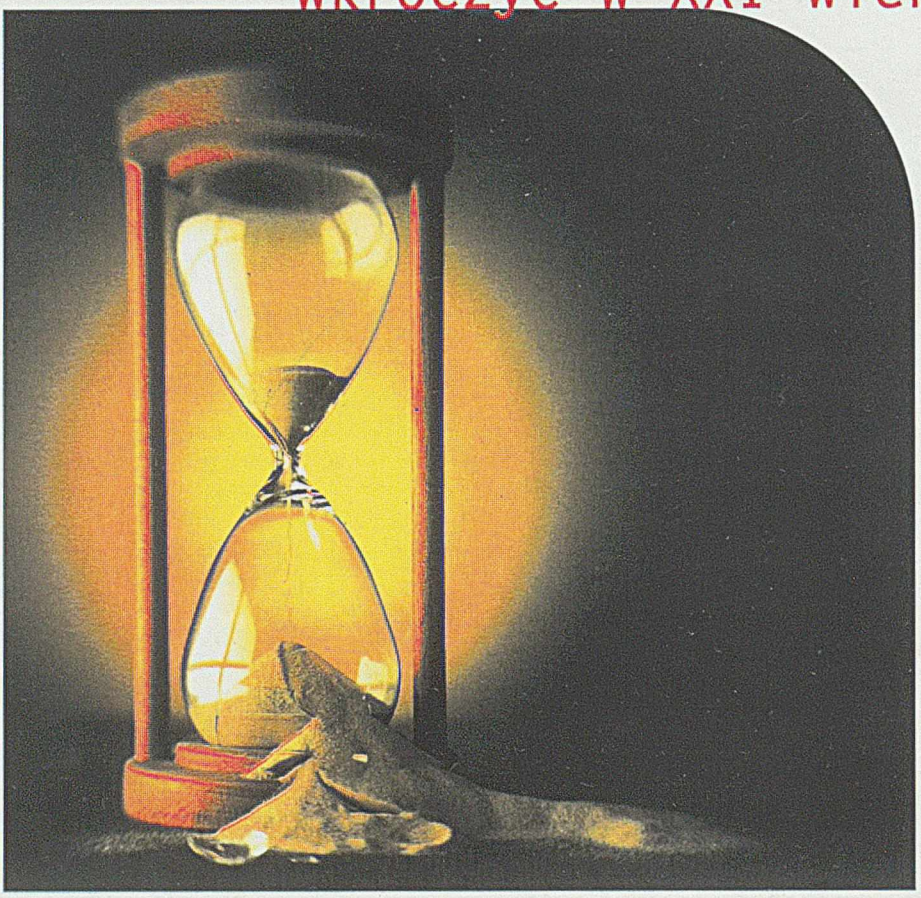
To gwarancja **jakości**, na którą Cię stać.

Polska OnLine
INTERNET SYSTEM INTEGRATOR

ul. Emaliowa 28, 02-295 Warszawa

tel. **(48-22) 868 0808**
fax **(48-22) 846 6271**
http://www.pol.pl
e-mail: **sales@pol.pl**

Czy twój system jest gotowy wkroczyć w XXI wiek?



Wraz z nadejściem roku 2000 użytkownicy wielu wykorzystywanych obecnie programów będą musieli sprostać nowym problemom. Dwie cyfry oznaczające rok będą wówczas cofały przedsiębiorstwo do roku 1900, zamiast umożliwić mu wkroczenie w następne tysiąclecie. Problem ten pojawił się już w przypadku długoterminowych planów i kontraktów. A czas ucieka.

System R/3® firmy SAP® umożliwia rozwiązanie tego i innych problemów, które mogą pojawić się w przyszłości. Dostępne są w nim na przykład funkcje przeliczania i obsługi wielu walut (związane z planowanym wprowadzeniem w całej Europie wspólnej waluty – Euro), a także aplikacje Internetowe, dzięki którym możliwe jest przeprowadzanie transakcji elektronicznych z wykorzystaniem międzynarodowej sieci komputerowej. Twórcy systemu R/3 zawsze spoglądają w przyszłość, oferując jeszcze więcej.

Już po krótkim etapie wdrożenia standardowy system R/3 wspomagający zarządzanie umożliwia automatyzację najważniejszych procesów realizowanych we wszystkich działach przedsiębiorstwa i powiązanie ich w sieć wzajemnych zależności. Dzięki takiemu połączeniu działów rachunkowości, produkcji, sprzedaży, zarządzania kadrami oraz innych obszarów przedsiębiorstwa uzyskuje się lepszy i pełniejszy przepływ informacji, co ma ogromny wpływ na szybkość podejmowania decyzji i ich trafność. Przedsiębiorstwo zaczyna funkcjonować w taki sposób, jakiego zawsze oczekiwało kierownictwo.

Dzięki modułowej budowie systemu R/3 każdy klient otrzymuje rozwiązanie idealnie dostosowane do swoich potrzeb. W dowolnym momencie system może być rozszerzany o kolejne aplikacje, dzięki czemu staje się najlepszym rozwiązaniem dla przedsiębiorstw o dowolnej wielkości, działających w każdym sektorze gospodarki.

Nasz adres:
SAP Polska Sp. z o.o. Mokotów Business Park
02-672 Warszawa, ul. Domaniewska 41
tel. (+48 22) 606 06 06; fax (+48 22) 606 06 07

SAP
A Better Return On Information.